The Visual Computer (2023) 39:6041–6072
https://doi.org/10.1007/s00371-022-02711-y

ORIGINAL ARTICLE

Fast colored video encryption using block scrambling and multi-key generation

Khalid M. Hosny[1] · Mohamed A. Zaki[1] · Nabil A. Lashin[1] · Hanaa M. Hamza[1]

Accepted: 20 October 2022 / Published online: 18 November 2022

Abstract

Multimedia information usage is increasing with new technologies such as the Internet of things (IoT), cloud computing, and big data processing. Video is one of the most widely used types of multimedia. Videos are played and transmitted over different networks in many IoT applications. Consequently, securing videos during transmission over various networks is necessary to prevent unauthorized access to the video's content. The existing securing schemes have limitations in terms of high resource consumption and high processing time, which are not liable to IoT devices with limited resources in terms of processor size, memory, time, and power consumption. This paper proposed a new encryption scheme for securing the colored videos. The video frames are extracted, and then, the frame components (red, green, and blue) are separated and padded by zero. Then, every frame component (channel) is split into blocks of different sizes. Then, the scrambled blocks of a component are obtained by applying a zigzag scan, rotating the blocks, and randomly changing the blocks' arrangements. Finally, a secret key produced from a chaotic logistic map is used to encrypt the scrambled frame component. Security analysis and time complexity are used to evaluate the efficiency of the proposed scheme in encrypting the colored videos. The results reveal that the proposed scheme has high-level security and encryption efficiency. Finally, a comparison between the proposed scheme and existing schemes is performed. The results confirmed that the proposed scheme has additional encryption efficiency.

Keywords Video encryption · IoT · Chaotic logistic map · Cryptography

1 Introduction

With the fast evolution of network technology and multimedia applications, video applications such as video-on-demand (VOD), video meetings, pay-tv, and video surveillance have been widely used. Because the video transmission depends on different networks, the video content may be captured because of the anatomy of the public channels. Securing colored videos during transmission and storage has become a challenging topic in recent years. The general video security objectives are availability, integrity, and confidentiality [1]. In general, three methods, i.e., video encryption (cryptography) [2–4], video steganography [5–8], and video watermarking [9, 10] could be used to achieve security. Cryptography is the most efficient technique to provide security to the colored videos by converting the raw video into an unintelligible video form using a secret key. The plain video can be restored only with the knowledge of the secret key. Video encryption techniques use two building blocks proposed by Shannon diffusion and confusion [11].

In general, image and video encryption algorithms are divided into full encryption and compression-combined encryption (selective encryption) [12]. Each of them has advantages and limitations. In full encryption [13–19], the whole image or video content is encrypted with a novel method directly, as shown in Fig. 1b. The full encryption algorithms are applied to uncompressed or compressed videos using any compression method [20]. The full encryption algorithms provide high-security encryption but take a long processing time. They are used in significant applications such as military and medical applications. In selective encryption [21–23], the video data are partially distorted by the encryption process, and the encrypted video is still partially intelligible after the encryption, as shown in Fig. 1c. They are used in applications that require low processing time. The proposed scheme wants to combine the advantages

✉ Khalid M. Hosny
k_hosny@yahoo.com; k_hosny@zu.edu.eg

[1] Department of Information Technology, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt

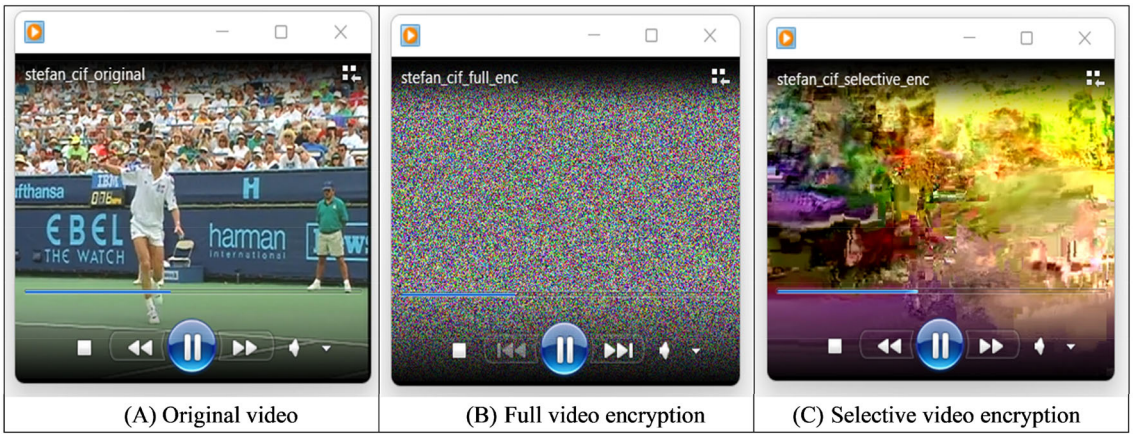


Fig. 1 Full and selective encryption methods

of the two mentioned methods to achieve good encryption and low processing time.

Because of the high correlation of the video frame neighboring pixels and the strong relationship between the video frames, traditional algorithms such as AES and DES could not guarantee high performance and low time processing for video encryption. The AES and DES are also unsuitable for encrypting colored video in real time [24]. Therefore, several algorithms for multimedia encryption were proposed [25–34]. These algorithms introduced by several academicians and researchers use different techniques such as DNA encoding and chaotic maps to encrypt images and videos securely and robustly. The most recent multimedia encryption methods are summarized in this section.

In [13], Li et al. presented a video encryption scheme that uses different chaotic algorithms and depends on the amount of information in each channel of a video frame. The video file is divided into a video stream and an audio stream. The video file stream is converted into YCbCr color space. The Arnold map and DNA encoding algorithm encrypt the Y channel, and the Lorenz hyperchaotic map is used to encrypt the other channels, where this scheme requires high-time processing. Yasser et al. proposed a multimedia encryption scheme based on hybrid-chaotic [19]. The proposed cryptosystem includes different media types such as videos, images, speech, and text. Alarifi et al. [16] developed a new hybrid cryptosystem for compressed video files based on chaotic maps, DNA sequences, and a modified Mandelbrot set. The scheme uses the Arnold map to generate three keys, and then, the encoding of the keys is performed with DNA sequences. The Hamming distance between the keys and a compressed YCbCr video frame is applied, encoding the result, and confusion and diffusion principles are applied. Valli and Ganesan[35] implemented a video encryption system that uses a substitution box to achieve diffusion and uses two different schemes. The first scheme is the higher-dimensional 12D chaos structure, and the other uses the Ikeda delay differential equation. The proposed drawbacks are the complexity of the key and the time that the encryption process takes. Kumar et al. [36] suggested a secure scheme based on chaos for video encryption. The algorithm provides a three-level of security: random selection of the frame, permutation order of the frame, and diffusion of the frame. In [37], Song et al. proposed a secure scheme to encrypt quantum videos. The proposed consists of three steps. First, permutation of the inter-frame position based on keys generated from an improved logistic map. Second, geometric transformation and improved logistic map change intra-frame pixels position. Finally, the quantum controlled-XOR operations and improved logistic map were used to encrypt the high 4-intra-frame-qubit-planes. In [38], Ye et al. used frequency domain encryption. First, the original image is transformed with the discrete wavelet transform and then compressed. Then, the carrier image is processed by lifting the wavelet transform and discrete cosine transform together with a Schur decomposition. Visually meaningful image encryption is achieved by embedding operation at the end. The encryption in the frequency domain improves encryption efficiency, but the implementation of frequency domain transformation leads to data loss. In [39], each channel of the color image was encrypted by the multi-parameter fractional discrete Tchebyshev moments. In [40], Gong et al. studied four-dimensional chaotic systems for image encryption applications. A new opto-digital color picture encryption scheme based on a compound chaotic map, the reality-preserving fractional Hartley transformation, and the piecewise linear chaotic map for image pixel replacement, optical processing, and permutation is suggested [41]. The proposed technique has a high sensitivity to keys and greater protection.

An overview of different schemes for securing colored video is introduced. Still, they have some drawbacks and

vulnerabilities: (1) the running time of the related algorithms is high and does not meet the real-time applications. (2) Some related algorithms are complex and unsuitable for IoT devices. (3) Some related algorithms evaluate their proposed work based on test images and do not investigate the test videos. (4) Some related algorithms do not investigate the effect of different noises in the security performance analysis. Motivated by previous points, this paper introduces a new scheme for securing the colored video with high-quality encryption to improve such shortcomings. The proposed scheme consists of a video preprocessing step plus four main steps: colored video components extraction and padding, frame components splitting, frame components scrambling, key generation, and diffusion step. The input-colored video is preprocessed to extract individual frames. The three video components (channels), red, green, and blue, are separated from each frame and padded by zeros. The four main steps are applied to each frame channel independently. First, the plain video frame channel is split into blocks, and the blocks are further split into sub-blocks by applying a new frame channel dividing scheme. Second, a scrambled frame channel is obtained by applying a zigzag scan in the blocks and the sub-blocks; then, a counterclockwise rotation by a 90° is applied to all blocks, and then, the blocks are shuffled randomly. Third, a key is generated based on the logistic map. Finally, the encrypted frame channel is obtained by applying the XOR function between the generated key and the scrambled frame channel.

The paper's contributions are summarized as follows:

1. A novel splitting method is introduced for each frame channel.
2. Random shuffling is performed between blocks to get a scrambled frame channel.
3. Diffuse the scrambled component using the logistic map, where the initial value of the logistic map is based on the first input frame component, making the proposed method robust against differential attacks.
4. The results show that the proposed scheme takes low processing time to encrypt the colored videos compared to the literature.

The rest of this paper is coordinated as follows. The proposed scheme is demonstrated in Sect. 2 in detail. Section 3 presents the simulation results and security analysis. Eventually, the work is concluded in Sect. 4.

2 The proposed video encryption method

This section describes the proposed method in detail. The raw colored video is preprocessed and encrypted in an unintelligible format. The decryption process is applied to get the original colored video. Figure 2 shows an illustrative diagram of the total steps.

2.1 Preprocessing the video

A. *Video components extraction* the proposed method is applied to each frame channel independently, so the input colored video is preprocessed to extract individual frames. Then, the frame channels are separated from each frame.
B. *Frame components padding* the encryption and decryption process needs the input video frame's size to be multiple of the block size. So, after the frame components are separated, it is needed to pad them by zeros according to the size of these components.

2.2 Encryption process

Here, the proposed scheme for encrypting colored video consists of four phases. These phases are performed on each channel independently. In the first phase, channel splitting is performed. In the second phase, channel scrambling (permutation) is applied. Key streams are generated from the logistic map in the third phase. The channel diffusion process is performed in the last phase.

2.2.1 Channel splitting

A raw frame channel is partitioned into blocks of equal size. The block size dimensions that the users can select from and are suitable for the scheme are 16, 32, and 64. Then, a random vector with a length equal to the number of blocks is generated. The blocks are further partitioned into sub-blocks or kept without partition based on the generated vector.

2.2.2 Channel scrambling

The arrangements of the frame channel's pixels are changed in this phase as follows:

(a) The zigzag scan is used to permute the positions of the pixels in each block (undivided and subdivided blocks) of the divided channel.
(b) Each block (undivided block and subdivided block) is rotated by 90°.
(c) For every block in the divided channel, a random number is generated to create a vector r.
(d) Depending on the vector r, a random permutation between blocks is performed to obtain the permuted frame channel.

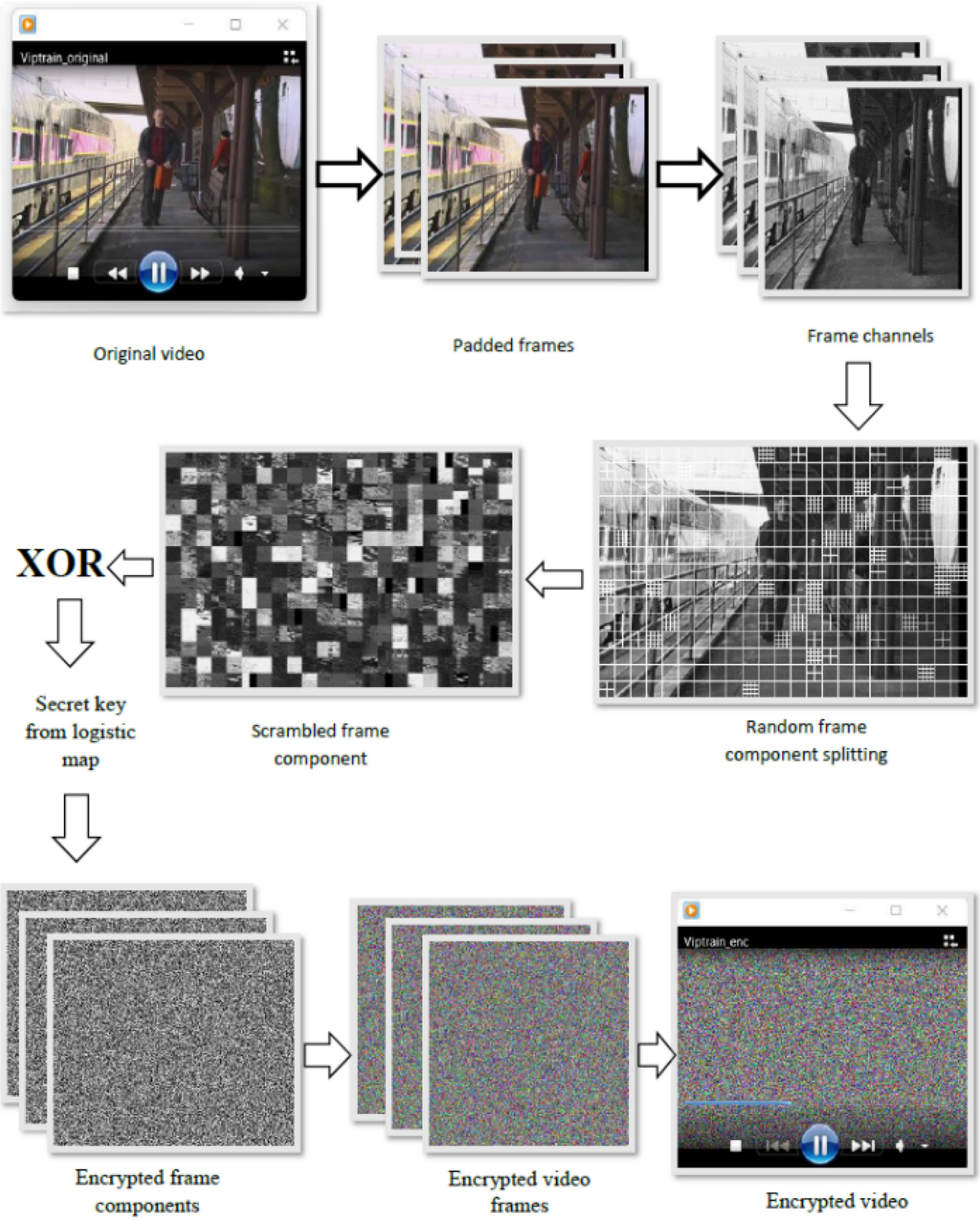


Fig. 2 Colored video encryption visual diagram

2.2.3 Key generation

A new key vector K from the logistic map is generated for every frame channel. The mathematical equation of the logistic map is:

$$Y_{n+1} = bY_n(1 - Y_n) \tag{1}$$

where $0 < b \leq 4$, and a starting value $0 < Y_0 < 1$. When $b \in [3.57, 4]$, the map is chaotic. The starting value Y_0 depends on the input colored video. The key generation steps for every frame channel are:

(a) The starting value of the logistic map is computed.

- For the first key vector (for the first channel of the first frame), Y_0 is calculated by:

$$Y_0 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} C(i,\ j)}{M \times N \times 255 \times 3} + 10^{-20} \tag{2}$$

where C is the input frame channel, and M and N are the input size.

- For other key vectors (for the other channels in the same frame or other frames), Y_0 value equals the last value of the previous key vector $K(MN)$ (in the previously processed channel).

(b) Get a sequence S_{temp} by iterating Eq. (1) $N_0 + MN$ times, then generate a new sequence S with size MN by discarding the first N_0 values of S_{temp}.

(c) Generate the key vector K by equation (3):

Algorithm 1. The steps of the proposed scheme for the colored video encryption process.

Input: the original colored video V, where M and N are the frame rows and columns, respectively, the parameter of the logistic map b, and the iterations number N_0.

1. Decomposition V into frames F, w is the number of extracted frames
2. Create a matrix V' to store the encrypted video frames
3. For i = 1 to w
4. Extract the components C of $F(i)$, l is the number of components
5. Create a matrix F' to store the encrypted components of a frame
6. For $j = 1\ to\ l$
7. Padding $C(j)$ if needed to get a padded frame component $C_p(j)$.
8. Split $C_p(j)$ into blocks, with a dimension size $g = 2^n$, where $n \in [4,5,6]$
9. Create a random sequence $R, with\ length\ (MN/g^2)$, where the random value = $2^u, where\ 2 \leq u \leq n$.
10. For jj =1 to length(R)
11. Depending on R_{jj}, $block_{jj}(C_p(j))$ is further partitioned into sub-blocks or is kept without partition.
12. End for
13. Apply a zigzag scan to all blocks.
14. Rotate all blocks by 90°.
15. Create a random sequence $R, with\ length\ (MN/g^2)$,
16. Depending on the sequence R, a random shuffle between the blocks of $C_p(j)$ is performed to obtain a permuted frame component P.
17. If i = 1 & j = 1
18. Compute the starting value of the logistic map by equation (2).
19. Else
20. The starting value of the logistic map equals the previous $K(MN)$.
21. End if
22. Iterate the equation (1) $N_0 + MN$ times, and store only the last MN elements in S.
23. For jj = 1 to MN
24. K(jj) = mod(floor(S(jj) × 10^{14}), 256).
25. End for
26. Generate $P' = reshape(P, 1, MN)$.
27. X = P′ ⊕ K
29. Generate $C'(j) = reshape(X, M, N)$.
29. F`(:,:,j) = C`(j).
30. End for
31. V`(i) = F`.
32. End for

Output: the encrypted video V`

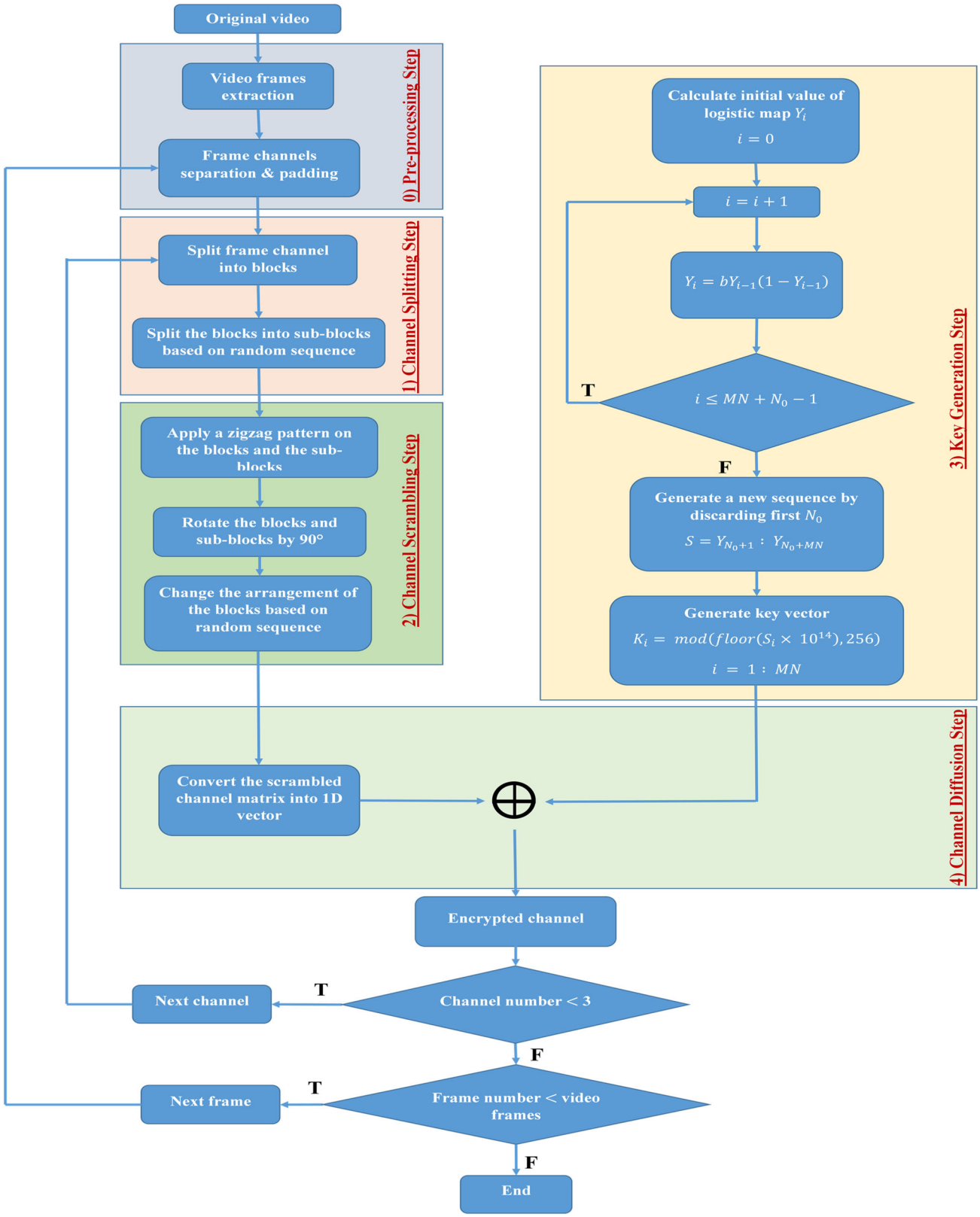


Fig. 3 Flowchart of the proposed scheme

$$K(i) = mod\Big(floor\Big(S(i) \times 10^{14}\Big), 256\Big), \quad i = 1 \text{ to} MN \tag{3}$$

2.2.4 Channel diffusion

In this phase, a bit-wise exclusive OR function is applied between every value in the generated key vector and the corresponding value in the permuted frame channel vector. After the channel pixels values are changed, an encrypted frame channel is generated. Algorithm 1 presents the steps of the encryption process. Also, Fig. 3 shows the flowchart of the scheme phases.

2.3 Decryption process

The decryption process can be constructed by inverting the encryption phases with the original keys to get the plain channels of each frame. The decryption steps are:

(1) The bit-wise exclusive OR function is performed between every value in the key vector and the corresponding value in the encrypted frame channel vector.
(2) Reordering the channel blocks placements to their original placements based on the random vector.
(3) Apply a rotation by -90° and inverse zigzag pattern to all blocks to rearrange the original placements of the pixels.

3 Simulation results and security analysis

This section examines the colored video encryption scheme for privacy and robustness. The colored videos used for testing are Train.avi (192 × 352 × 3), Rhinos.avi (240 × 320 × 3), Viptrain.avi (240 × 360 × 3) and Flamingo.avi (192 × 352 × 3) taken from Valli and Ganesan [35], and Foreman.avi (352 x 288 x 3) downloaded from YUV Sequences [42]. Figure 4 shows the test video samples. The proposed scheme is executed using MATLAB (R2015a) on a laptop that has the subsequent specifications: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz, 16 GB memory, and Windows 11 OS. The algorithm's initial parameters are: In the channel splitting step, the dimension of the blocks is 16 (where $n = 4$), $b = 3.9$ for the logistic map, and $N_0 = 1000$ for the skipped elements.

3.1 Visual analysis

Different evaluation metrics have been used with the proposed scheme. The first metric used to evaluate the scheme is the visual inspection. The encryption/decryption results of the videos are displayed in Fig. 4. The results indicate that the scheme hides all details within the test videos, and the receiver side restores the original videos successfully.

3.2 Histogram analysis

A histogram is an essential tool in evaluating the efficiency of the encryption scheme. It represents the number of occurrences of each pixel value in a frame channel. The flat histogram indicates that the frame channel can resist different types of statistical attacks [43]. Figures 5, 6, 7, 8 and 9 show the histograms for various videos' 10th original, encrypted, and decrypted frames. It is observed that the encrypted frames histograms have a uniform distribution form and are not similar to their corresponding original frames histograms.

Consequently, the proposed scheme hides any pattern in the frames of the test videos. Additionally, the decrypted frames histograms and their corresponding original frames are the same. So, the scheme can recover the original frame from the encrypted one successfully.

3.3 Correlation analysis

Principally in each video frame, there is a high correlation between neighboring pixels as the intensity values are nearly the same. These relationships must be reduced to protect the video frame against different attacks. The adjacent pixels pair's correlation can be calculated using the following equations.

$$r_{A,B} = \frac{E((A - E(A))(B - E(B)))}{\sqrt{D(A)D(B)}} \tag{4}$$

$$E(A) = \frac{1}{s}\sum_{i=1}^{s} A_i \tag{5}$$

$$D(A) = \frac{1}{s}\sum_{i=1}^{s}(A_i - E(A))^2 \tag{6}$$

where A and B represent the two adjacent pixel values, and s is the total number of selected pairs. Figures 10, 11 and 12 show the horizontal (H), vertical (V), and diagonal (D) correlation distributions of 6000 random pairs of neighboring pixels selected for the 10th original and encrypted frame of the Flamingo test video. The correlation values of 6000 random pairs of adjacent pixels for the 10th original and encrypted frame of various videos, along with H, V, and D directions, are presented in Table 1. From the results, the values of the original frames are close to one. On the contrary, the values of the encrypted frames are very low and very close to zero. So, there is no correlation between pixels

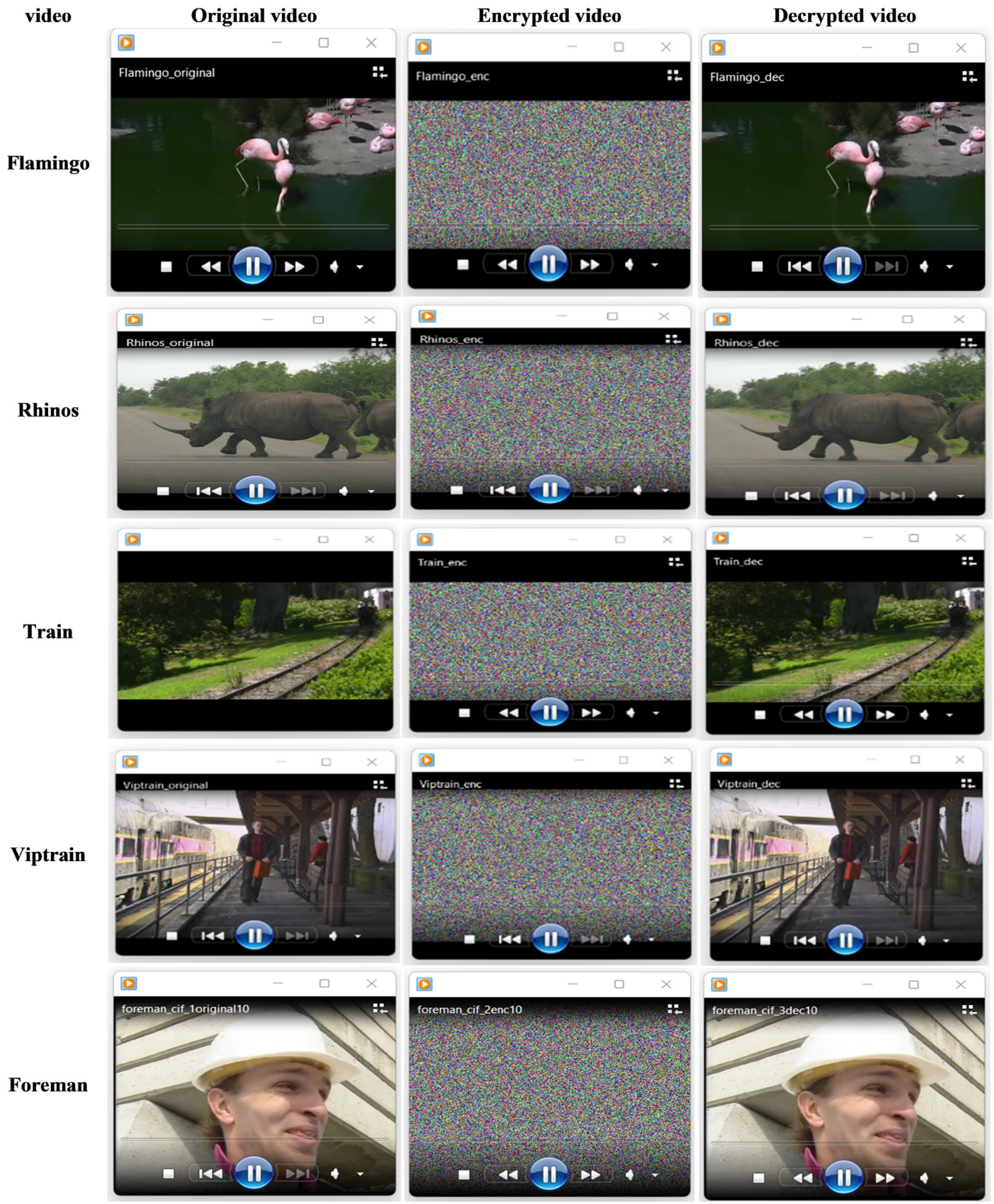


Fig. 4 Original, encrypted, and decrypted videos

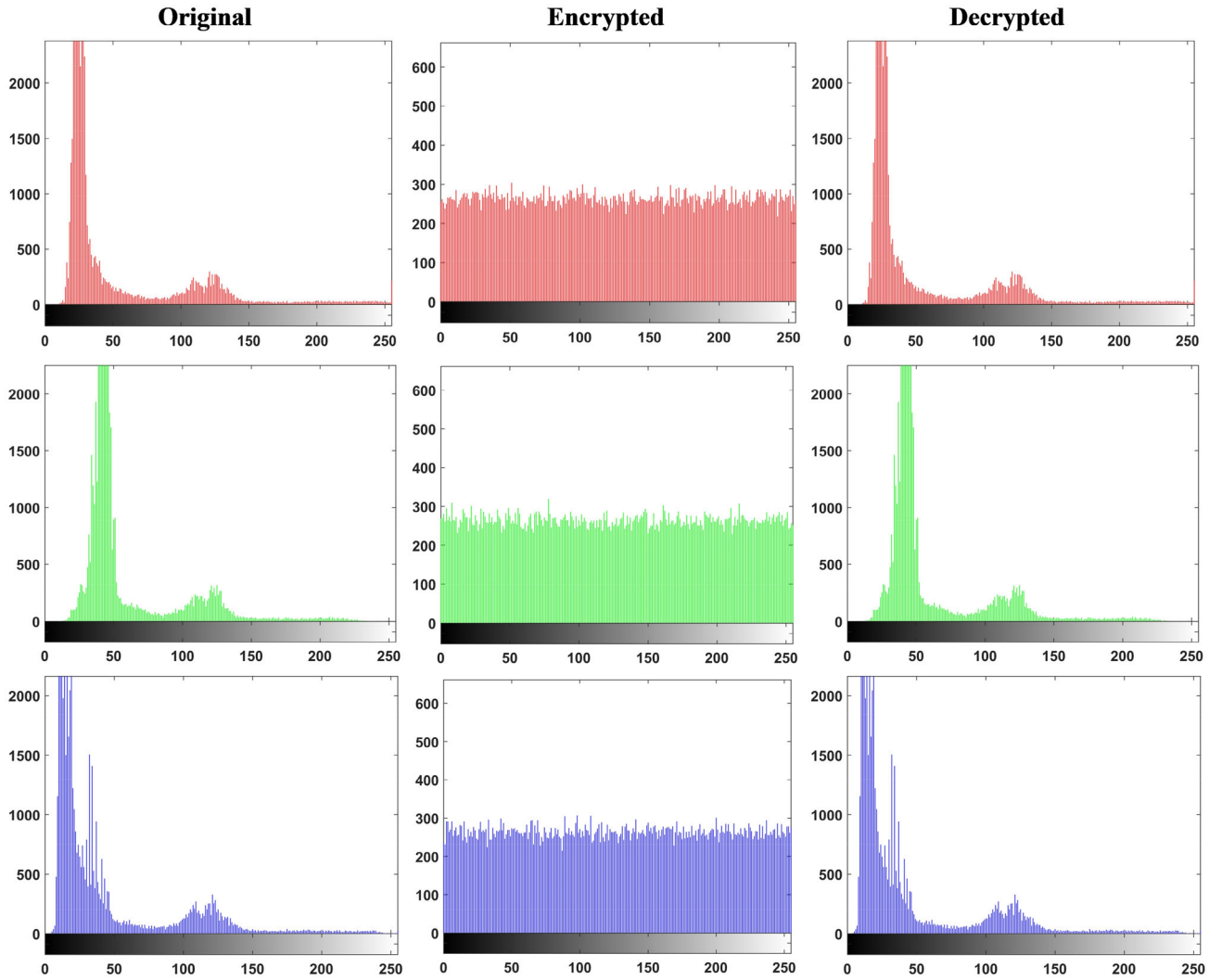


Fig. 5 Histogram for Flamingo video

in the frames encrypted by the proposed scheme. Therefore, the proposed scheme can resist statistical attacks.

3.4 Entropy analysis

The information entropy is used to measure the randomness of the video frames. The Shannon entropy defines the degree of randomness of a video frame. The mathematical definition of entropy is calculated by

$$H(m) = \sum_{i=1}^{w} p(m_i) \log_2 \frac{1}{p(m_i)} \tag{7}$$

where the m_i represents the ith gray value in a video frame, and $p(m_i)$ is the probability of m_i in a video frame. To ensure the randomness of the encrypted video frame with the suggested scheme, the entropy value of the encrypted frame should be near 8. The entropy values for the 10th frame of various videos are presented in Table 2. From the table, all values are close to 8, which indicates that the videos protected by the proposed scheme are robust against entropy attacks.

3.5 Differential attack

An adversary can conjecture information about the video frame by changing an original video frame and then encrypting the original video frame and the modified original video frame using the same encryption method. The adversary compares the two encrypted frames with the plain frame and searches for the relationships between them. Therefore, the encryption scheme should generate a different encrypted frame with every little change in the original. The metrics used to evaluate algorithm performance for this aim are NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity). The mathematical calculations

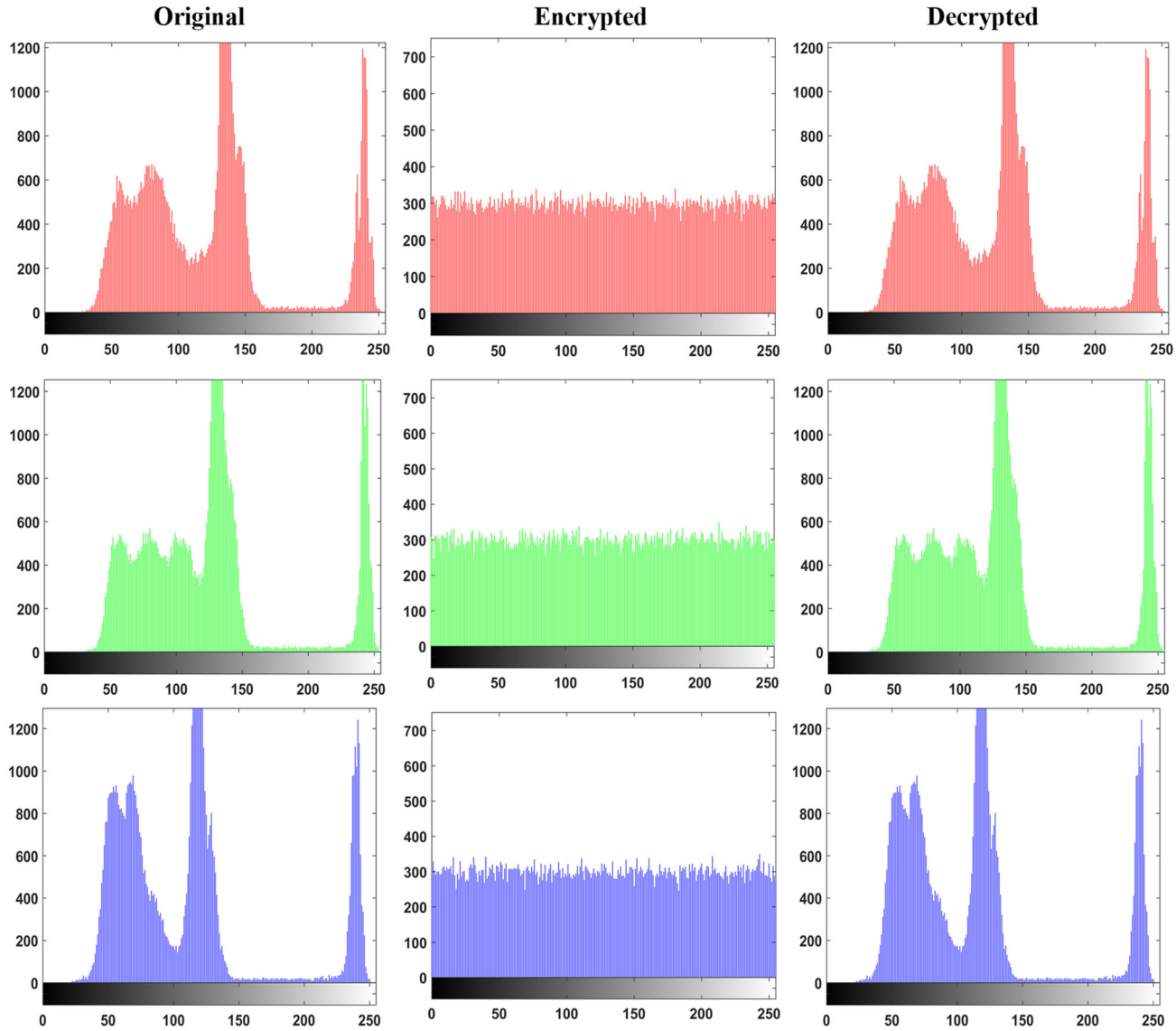


Fig. 6 Histogram for Rhinos video

for the metrics are:

$$\text{NPCR} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,\ j) \times 100(\%)$$

$$D(i,\ j) = \begin{cases} 0 \text{ if } C_1(i,\ j) = C_2(i,\ j), \\ 1 \text{ if } C_1(i,\ j) \neq C_2(i,\ j), \end{cases} \tag{8}$$

$$\text{UACI} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{|C_1(i,\ j) - C_2(i,\ j)|}{255} \times 100(\%) \tag{9}$$

where C_1 and C_2 are the encrypted video frame (plain and modified video frames). The modified frame is made by changing one pixel in the plain video frame. M and N are the video frame size. The ideal values for NPCR and UACI are 99.6094 and 33.4635%, respectively. The NPCR and UACI values for the proposed scheme applied on the 10th frame of various videos are presented in Table 3. From the table, NPCR and UACI are very close to the ideal values. Therefore, the videos encrypted by the proposed scheme have great resistance against differential attacks.

3.6 Encryption quality analysis

3.6.1 Histogram deviation (D_H)

A metric is used to evaluate the quality of encryption for the proposed scheme by measuring the deviation in pixels values between the original video frame and the encrypted one. The

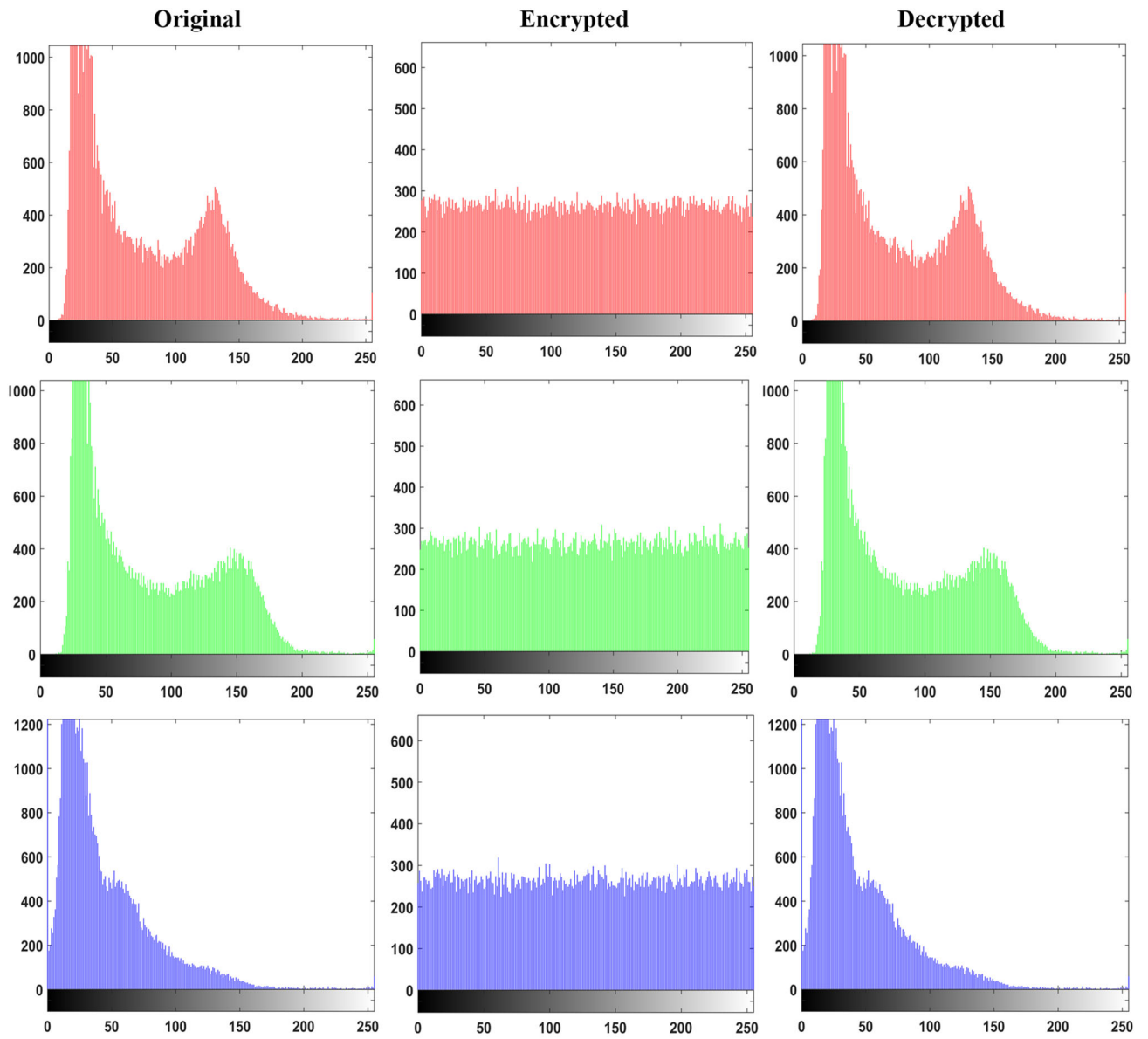


Fig. 7 Histogram for Train video

maximum deviation can be estimated by:

$$D_H = \frac{K_0 + K_{255}}{2} + \sum_{i=1}^{254} K_i \tag{10}$$

where K_i is the difference at gray value i. The large value of maximum deviation states a high deviation in the encrypted video frame from the original one. Table 4 presents the D_H values for the 10th original frame and encrypted frame for various videos using the proposed scheme. From the table, it is observed that the D_H values between the original and encrypted videos are large, proving that the quality of the videos encrypted by the proposed scheme is good enough.

3.6.2 Irregular deviation (D_I)

A metric used to measure the maximum irregular deviation quantity in an encrypted video frame caused by an encryption algorithm. The irregular deviation can be estimated by:

$$D_I = \frac{\sum_{i=0}^{255} |H(i) - A|}{M \times N} \tag{11}$$

where H_i refers to the histogram of the difference between the original and encrypted video frame at index i, and A, is the mean value of the histogram for the encrypted video frame. The lower value of D_I indicates that the pixel distribution is uniform, and the quality of the encrypted video is high. Table

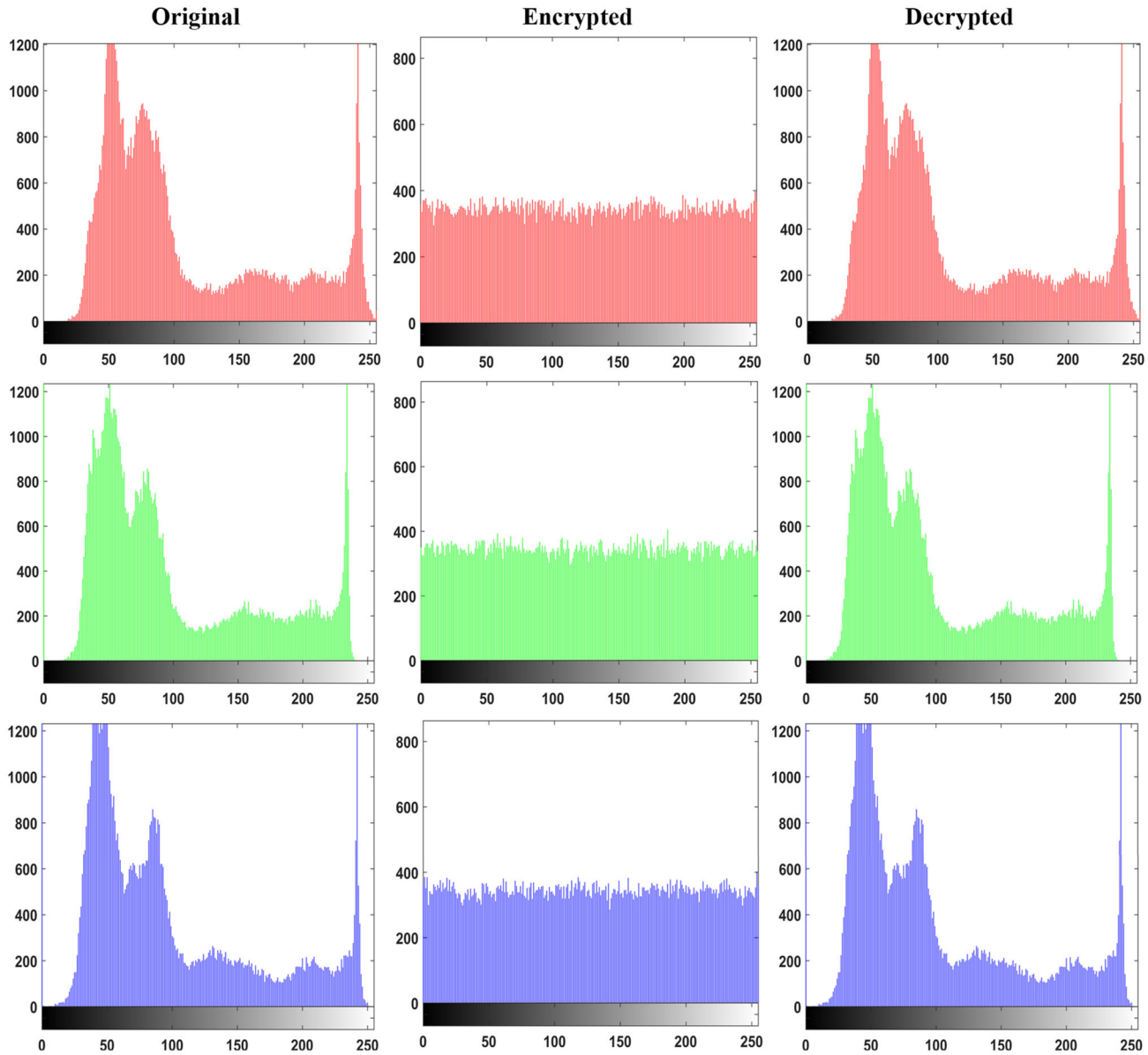


Fig. 8 Histogram for Viptrain video

4 presents the D_I values for the 10th original and encrypted frames for the various videos using the proposed scheme. The results in the table show that the D_I values are low, proving the high quality of the encrypted videos and hence the strength of the proposed scheme.

3.7 PSNR, SSIM, and FSIM analysis

The peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and feature similarity (FSIM) metrics are used to estimate the quality performance of the encryption and decryption processes. This experiment evaluates the PSNR, SSIM, and FSIM between the original and encrypted video frames. The encryption process is efficient if the result values are low. Also, the PSNR, SSIM, and FSIM are evaluated between the original and decrypted video frames. The decryption process is efficient if the result values are high.

(1) The PSNR measures the ratio between the highest possible power of a signal and the power of distorted noise. The PSNR for a grayscale video frame is measured by:

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{255 \times 255}{\mathrm{MSE}}\right)(\mathrm{dB})$$

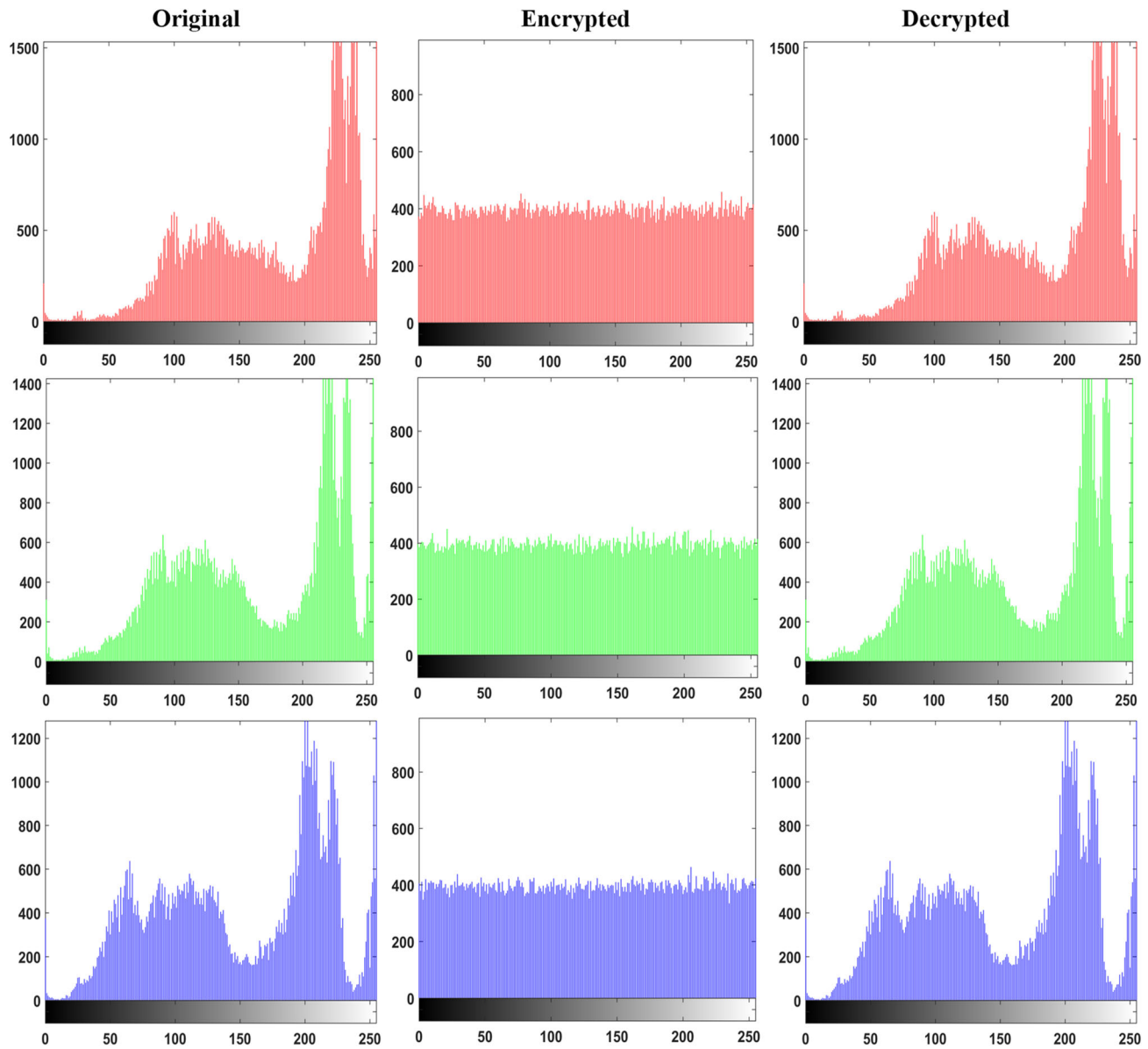


Fig. 9 Histogram for Foreman video

$$\text{MSE} = \frac{1}{MN} \times \sum_{i=1}^{m} \sum_{j=1}^{n} |F_1(i,\ j) - F_2(i,\ j)|^2$$

where F_1 is the original video frame component, while F_2 is the encrypted one. Small values of PSNR between the original video frame component and the corresponding encrypted one indicate a good encryption process. Table 5 shows the results of PSNR values between the 10th original and encrypted frames for various videos. The proposed scheme has low PSNR values from the table, indicating that the encryption process is efficient. Also, Table 6 shows the results of PSNR values between the 10th original and decrypted frames for various videos. The proposed scheme has high PSNR values from the table, indicating that the decryption process is efficient.

(2) The SSIM index measures the similarity between two video frames and ranges from − 1 to 1 decimal value. The SSIM value can be calculated using:

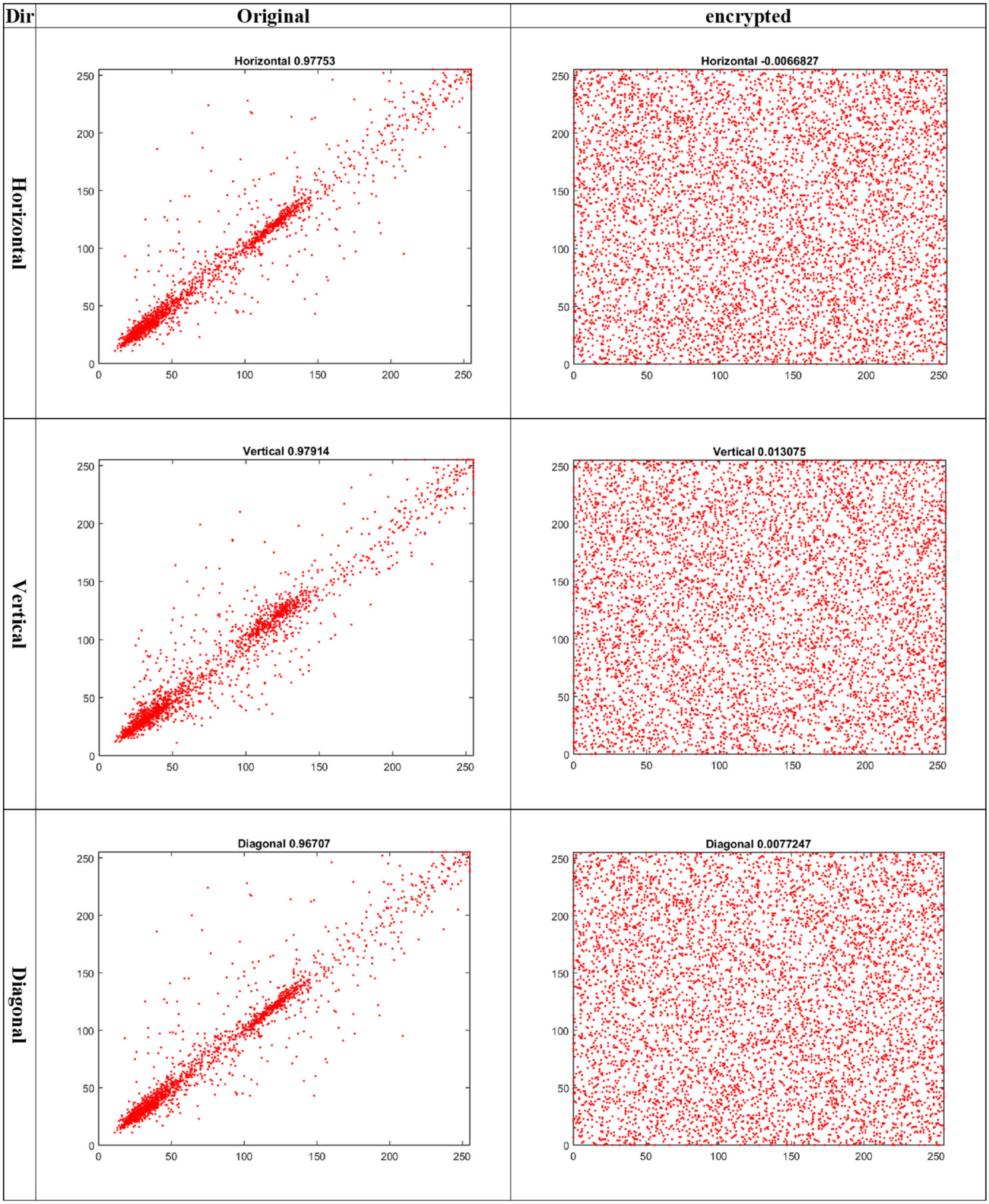


Fig. 10 Correlation distribution for the red channel of Flamingo video

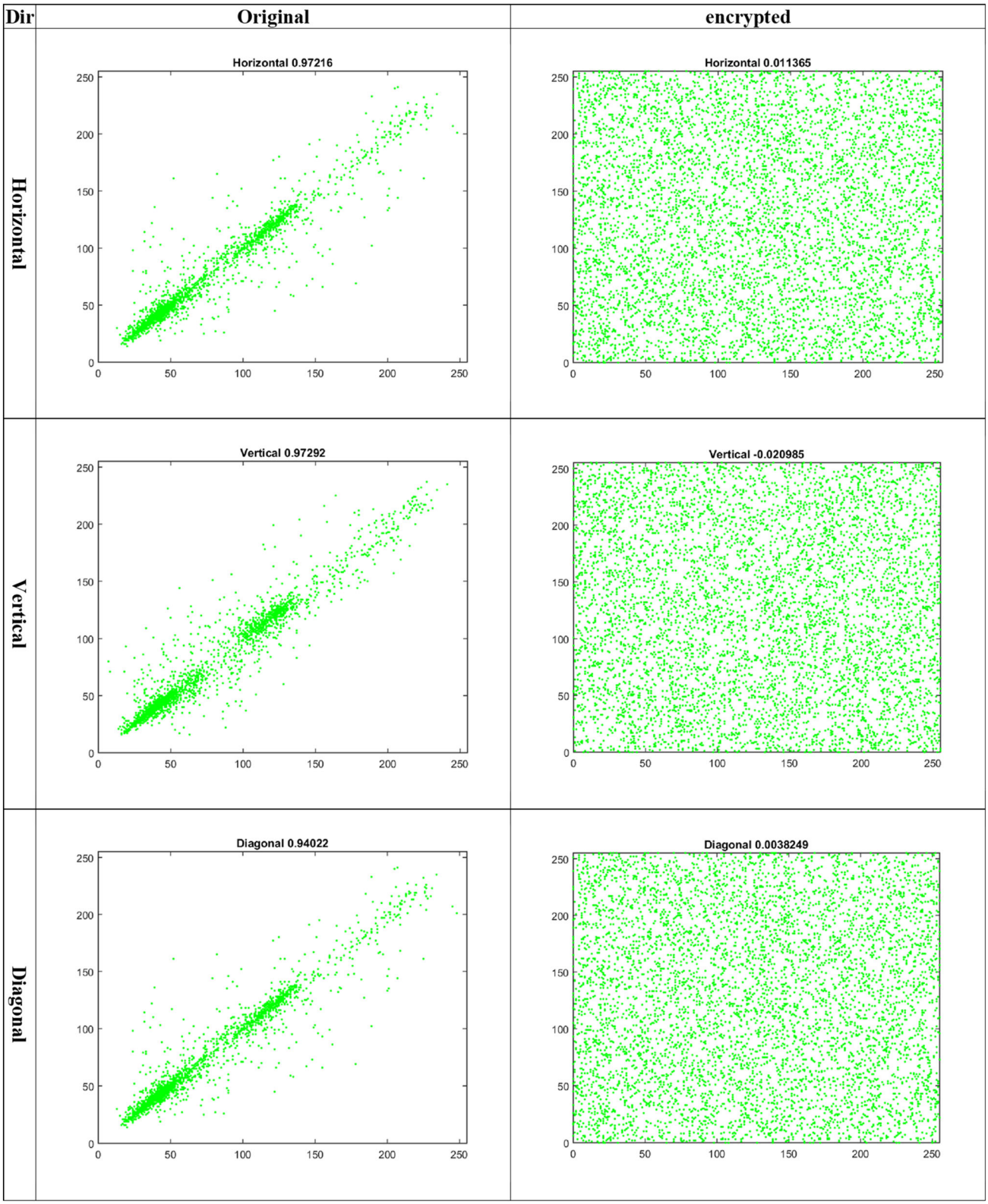


Fig. 11 Correlation distribution for the green channel of Flamingo video

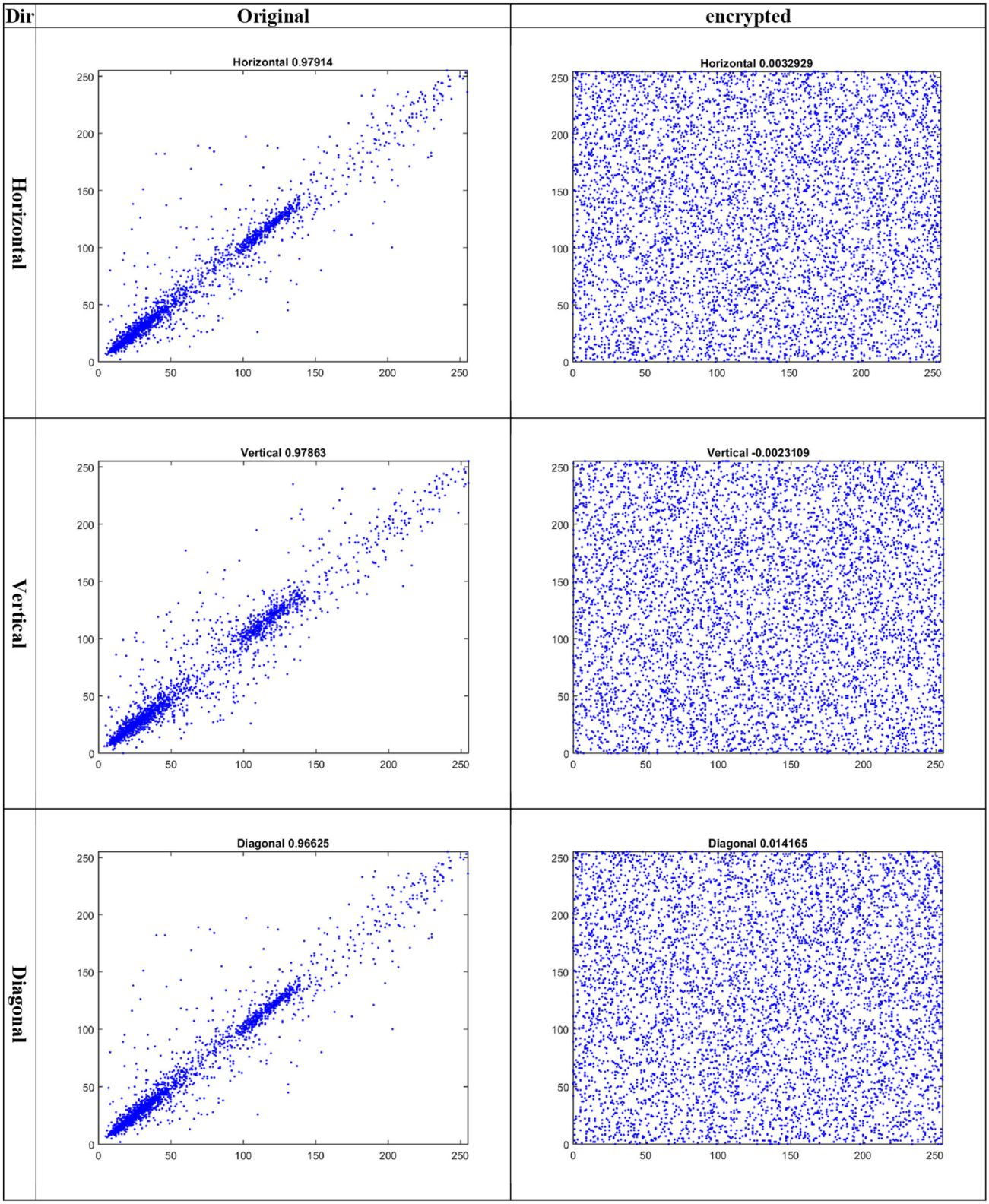


Fig. 12 Correlation distribution for the blue channel of Flamingo video

Table 1 Correlation coefficients for various videos

Video	Channel	Original frame			Encrypted frame		
		H	V	D	H	V	D
Flamingo	R	0.9775	0.9791	0.9671	− 0.0067	0.0131	0.0077
	G	0.9722	0.9729	0.9402	0.0114	− 0.0209	0.0038
	B	0.9791	0.9786	0.9663	0.0033	− 0.0023	0.0142
Rhinos	R	0.9882	0.9844	0.9782	− 0.0019	− 0.0025	− 0.0049
	G	0.9881	0.9845	0.9781	0.0005	− 0.0299	0.0032
	B	0.9899	0.9852	0.9776	− 0.0045	− 0.0066	− 0.0124
Train	R	0.9448	0.9190	0.8873	0.0210	− 0.0123	− 0.0114
	G	0.9469	0.9277	0.893	− 0.0257	0.0055	0.0203
	B	0.9028	0.8772	0.8218	− 0.0148	0.0197	0.0205
Viptrain	R	0.9512	0.9688	0.9146	− 0.0054	0.0037	− 0.0140
	G	0.9519	0.9696	0.9159	0.0069	− 0.0180	− 0.0037
	B	0.9491	0.9654	0.9179	0.0252	− 0.0018	− 0.0275
Foreman	R	0.9778	0.9702	0.9588	− 0.0213	− 0.0059	− 0.0028
	G	0.9839	0.9744	0.9662	0.0192	− 0.0094	0.0159
	B	0.9856	0.9789	0.9718	− 0.0276	0.0011	0.0216

Table 2 Entropy values for various videos

Video	Channel	Original frame	Encrypted frame	Decrypted frame
Flamingo	R	5.4895	7.9975	5.4895
	G	5.5577	7.9974	5.5577
	B	5.8021	7.9974	5.8021
Rhinos	R	6.9800	7.9978	6.9800
	G	6.9031	7.9975	6.9031
	B	6.7746	7.9975	6.7746
Train	R	7.1029	7.9975	7.1029
	G	7.1377	7.9970	7.1377
	B	6.7096	7.9974	6.7096
Viptrain	R	7.3747	7.9977	7.3747
	G	7.3450	7.9980	7.3450
	B	7.3925	7.9979	7.3925
Foreman	R	7.2887	7.9982	7.2887
	G	7.4253	7.9981	7.4253
	B	7.5765	7.9984	7.5765

$$\mathrm{SSIM}(x,\ y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}$$

where μ_x and μ_y, respectively, represent the average value of original frame x and encrypted frame y, σ_x^2 and σ_y^2, respectively, represent the corresponding variance value, σ_{xy} is the covariance of $xandy$, and $c1$ and $c2$ are constants. Table 5 presents the SSIM values between the 10th original and encrypted frames for various videos.

Table 6 presents the SSIM values between the 10th original and decrypted frames for various videos. Getting lower SSIM values between the original and encrypted frames is recommended to prove the encryption process's efficiency. It is recommended to get higher SSIM values between the original and decrypted frames to prove the efficiency of the decryption process. From Tables 5 and 6, the SSIM values between the original and encrypted frames are low, and the SSIM values between the original and decrypted frames are high, proving the quality of the encryption and decryption processes.

Table 3 The NPCR and UACI values for various videos.

Video	Channel	NPCR	UACI
Flamingo	R	99.5827	33.4939
	G	99.6094	33.5092
	B	99.5605	33.5853
Rhinos	R	99.6068	33.5630
	G	99.6237	33.3876
	B	99.6250	33.3666
Train	R	99.5694	33.3913
	G	99.6094	33.2731
	B	99.6212	33.5992
Viptrain	R	99.6286	33.5842
	G	99.5811	33.4908
	B	99.6445	33.4451
Foreman	R	99.6183	33.4292
	G	99.5778	33.5481
	B	99.5975	33.4006

Table 4 Histogram deviation and irregular deviation values for various videos

Video	Channel	D_H	D_I
Flamingo	R	89679.0	0.3079
	G	90577.0	0.4119
	B	87707.0	0.2588
Rhinos	R	72166.5	0.6583
	G	76752.5	0.6808
	B	84182.5	0.6163
Train	R	55991.5	0.4918
	G	51430.0	0.5299
	B	73583.0	0.3225
Viptrain	R	68693.0	0.5006
	G	69239.5	0.4923
	B	67440.0	0.4915
Foreman	R	67251.0	0.4692
	G	64334.5	0.4934
	B	55782.5	0.5223

(3) FSIM evaluates the local symmetry between the original and encrypted video frames. The FSIM value can be calculated using:

$$\text{FSIM} = \frac{\sum_{x\in\Omega} S_L(x)\cdot PC_m(x)}{\sum_{x\in\Omega} PC_m(x)}$$

where $S_L(x)$ represents the total anticipated similarity between two video frames, Ω is the spatial video frame, and $PC_m(x)$ is the congruency phase value. Table 5 presents the FSIM values between the 10th original and encrypted frames for various videos. Table 6 presents the FSIM values between the 10th original and decrypted frames for various videos. It is recommended to get lower FSIM values between the original and encrypted frames to prove the efficiency of the encryption process. It is recommended to get higher FSIM values between the original and decrypted frames to prove the efficiency of the decryption process. From Tables 5 and 6, the FSIM values between the original and encrypted frames are low, and the FSIM values between the original and decrypted frames are high, proving the quality of the encryption and decryption processes.

3.8 Chosen-plaintext and known-plaintext attacks analysis

In this section, the resistance of the proposed scheme against chosen-plaintext and known-plaintext attacks is tested. Two different videos are used in this experiment. The first video is white, and the second is a black video. The videos are encrypted using the proposed scheme. The original and encrypted videos are shown in Fig. 13. The encrypted videos have no valuable information. So the proposed scheme has higher robustness against chosen-plaintext and known-plaintext attacks. Also, Table 7 shows the entropy value of the original and encrypted videos. From the table, the entropy value of the encrypted videos is very close to the optimal value, reflecting the strength of the proposed scheme.

3.9 Edges detection analysis

The encryption scheme must guarantee to protect the edge information of the encrypted video. The edge differential ratio (EDR) metric is used in this experiment to estimate the edge distortion and is defined by:

$$EDR = \frac{\sum_{i,\,j=1}^{k} \left| P(i,\,j) - \overline{P}(i,\,j) \right|}{\sum_{i,\,j=1}^{k} \left| P(i,\,j) + \overline{P}(i,\,j) \right|}$$

Where the pixel values in the edges within the binary form of the original video and encrypted video are $P(i,\,j)$ and $\overline{P}(i,\,j)$, respectively. The EDR value should be close to one to ensure that the original and encrypted video is dissimilar. The EDR values between the 10th original and encrypted frames for various videos are presented in Table 8. From the table, the values are close to one, and the proposed scheme guarantees the original video and encrypted video are different. The Laplacian of Gaussian edge detection for the 10th original encrypted and decrypted frames for various videos is displayed in Fig. 14. The displayed results show a big difference between the original and encrypted frames on the

Table 5 PSNR, SSIM, and FSIM values between the original and encrypted frames of various videos

Video	Channel	PSNR (dB)	SSIM	FSIM
Flamingo	Red	6.59	0.0039	0.2011
	Green	7.43	0.0069	0.2117
	Blue	6.22	0.0049	0.2052
Rhinos	Red	8.77	0.0093	0.2545
	Green	8.81	0.0097	0.2477
	Blue	8.48	0.0087	0.2486
Train	Red	7.71	0.0066	0.4469
	Green	7.99	0.0080	0.4452
	Blue	6.69	0.0044	0.4578
Viptrain	Red	7.94	0.0097	0.3742
	Green	7.94	0.0077	0.3762
	Blue	7.89	0.0095	0.3798
Foreman	Red	7.45	0.0115	0.2969
	Green	7.66	0.0096	0.2903
	Blue	7.98	0.0096	0.2961

Table 6 PSNR, SSIM, and FSIM values between the original and decrypted frames of various videos

Video	Channel	PSNR (dB)	SSIM	FSIM
Flamingo	Red	Inf.	1	1
	Green	Inf.	1	1
	Blue	Inf.	1	1
Rhinos	Red	Inf.	1	1
	Green	Inf.	1	1
	Blue	Inf.	1	1
Train	Red	Inf.	1	1
	Green	Inf.	1	1
	Blue	Inf.	1	1
Viptrain	Red	Inf.	1	1
	Green	Inf.	1	1
	Blue	Inf.	1	1
Foreman	Red	Inf.	1	1
	Green	Inf.	1	1
	Blue	Inf.	1	1

edges. So the proposed scheme can hide the main details in the videos. Also, the edges in original frames are similar to those in decrypted frames, proving the proposed scheme's efficiency in decryption.

3.10 Keyspace analysis

The colored video encryption scheme should have a large keyspace to be robust and secure. The scheme can escape from the brute-force attacks if the keyspace $\geq 2^{100}$. The proposed scheme uses different initial values to generate the secret key: the starting value Y_0, and the control parameter b of the logistic map, and the number of skipped elements N_0. We consider the $Y_0 and b$ precision is 10^{16}, and N_0 precision is 10^3; therefore, 10^{35} is the total space of the key. So, the proposed scheme can withstand such attacks because the keyspace is larger than 2^{100}.

3.11 Key sensitivity analysis

Any slight modification to the secret key of the encryption scheme should generate considerable changes in the result. The adversary uses a similar secret key to break the encryption scheme in the decryption process. The test video frame is

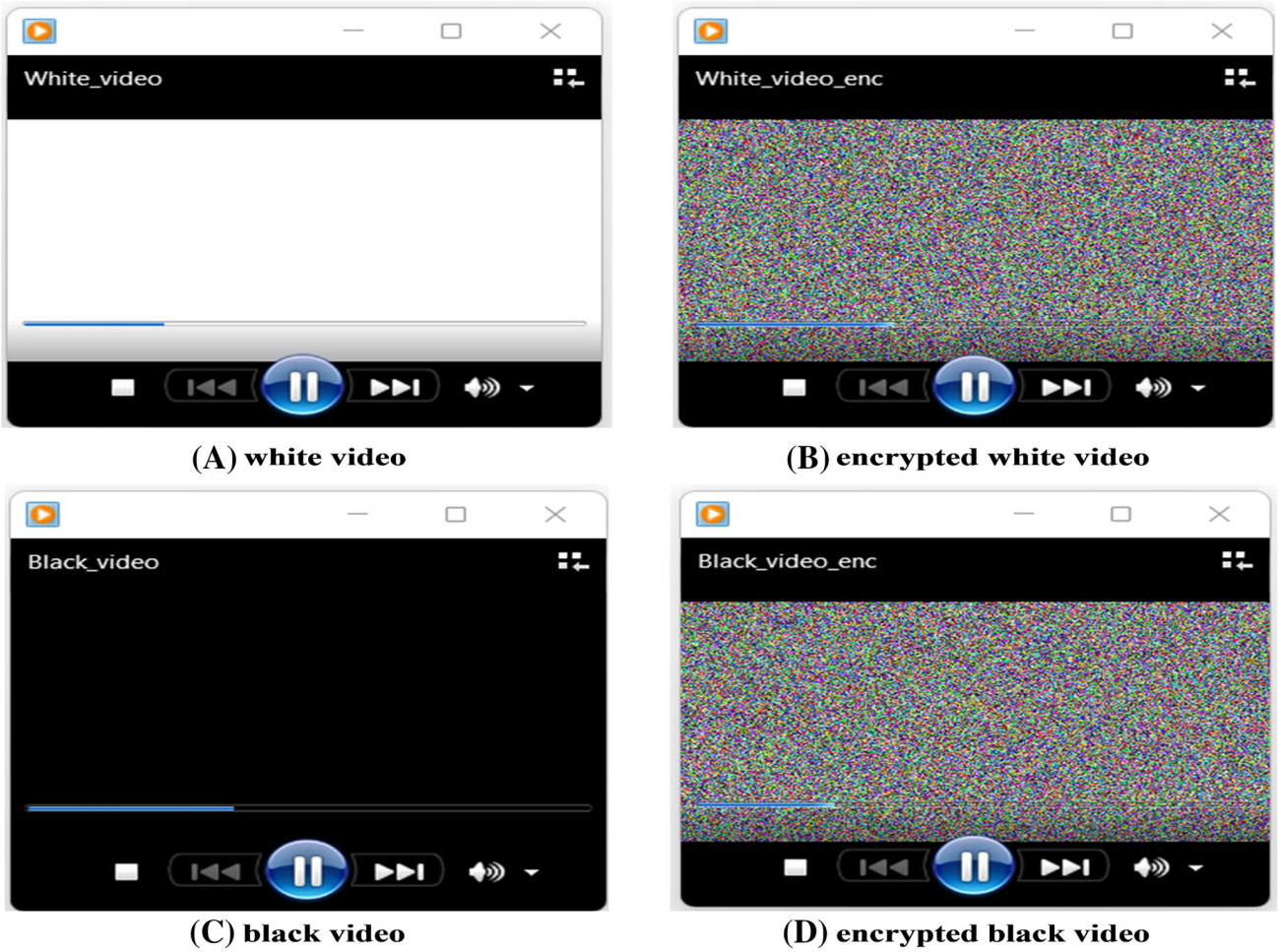


Fig. 13 Original and encrypted version of white and black videos

encrypted with a key 1 generated from the chaotic map, with starting value $Y_0 = z$ as shown in Fig. 15B to test the key sensitivity. Later the encrypted frame is decrypted twice: once with a slight change in the starting value $Y_0 = z + 10^{-10}$ as shown in Fig. 15C and again with key 1 as shown in Fig. 15D. It is concluded that only the same secret key used in the encryption process can restore the original frame in the decryption process, and any slight change in the secret key will fail to break the encryption scheme.

3.12 Channel noises attack analysis

After the video is encrypted, it can be transmitted through different communication channels. During the transmission, the encrypted video may be affected by some noise. So, different types of noises are used to prove the efficiency of the proposed scheme's decryption process.

Table 7 Entropy values for original and encrypted white and black videos

Video	Original video	Encrypted video
White video	0	7.9990
Black video	0	7.9992

3.12.1 Salt & peppers noise

In this experiment, the salt and pepper noise with variance value 0.005 is added to various encrypted video frames, and then, the decryption process is performed. The effect of this type on a video frame results in black and white dots on the video frame. Figure 16 shows that the decrypted videos are still intelligible, despite the effect of the noise on the video frames, proving the proposed scheme's power.

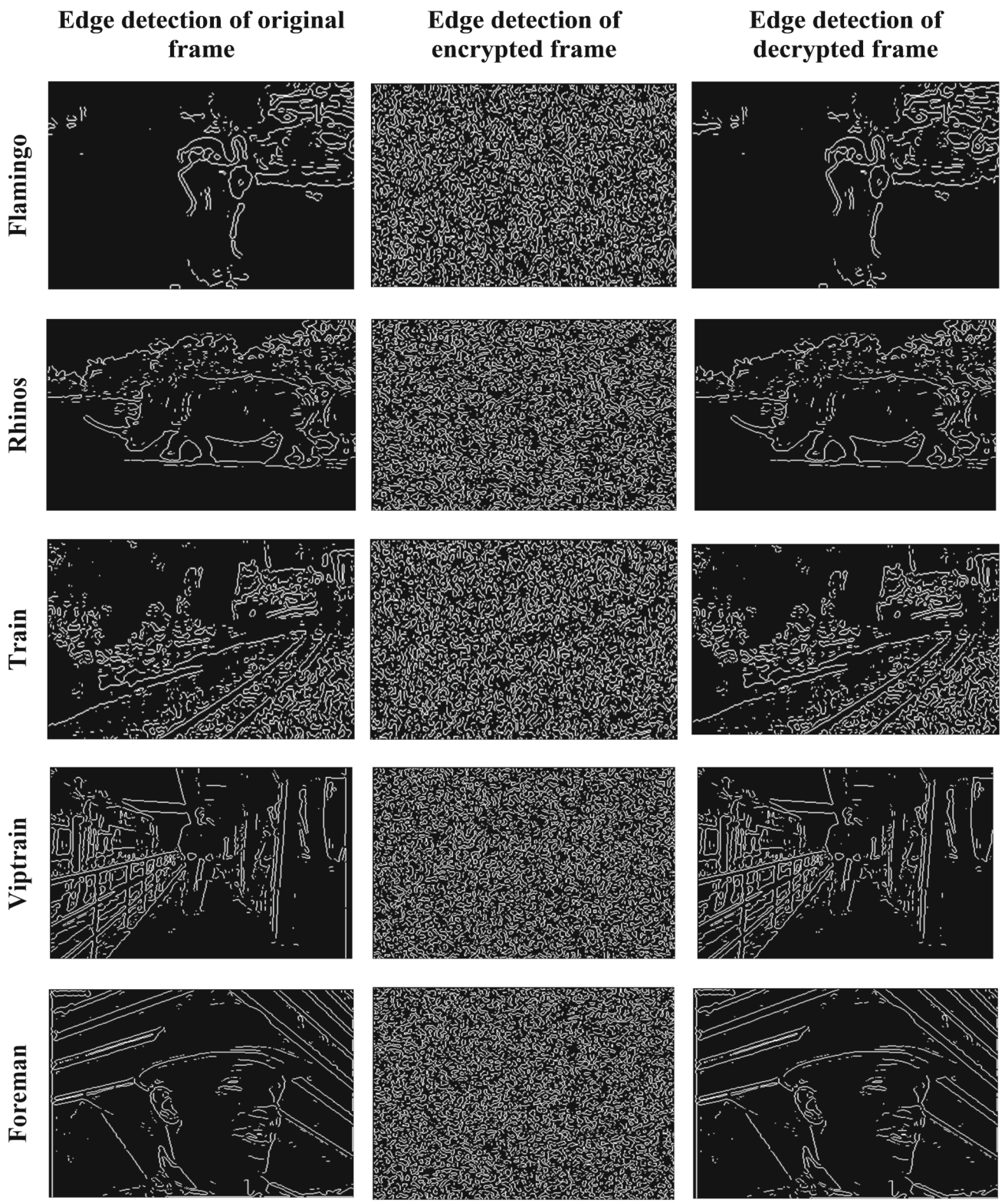
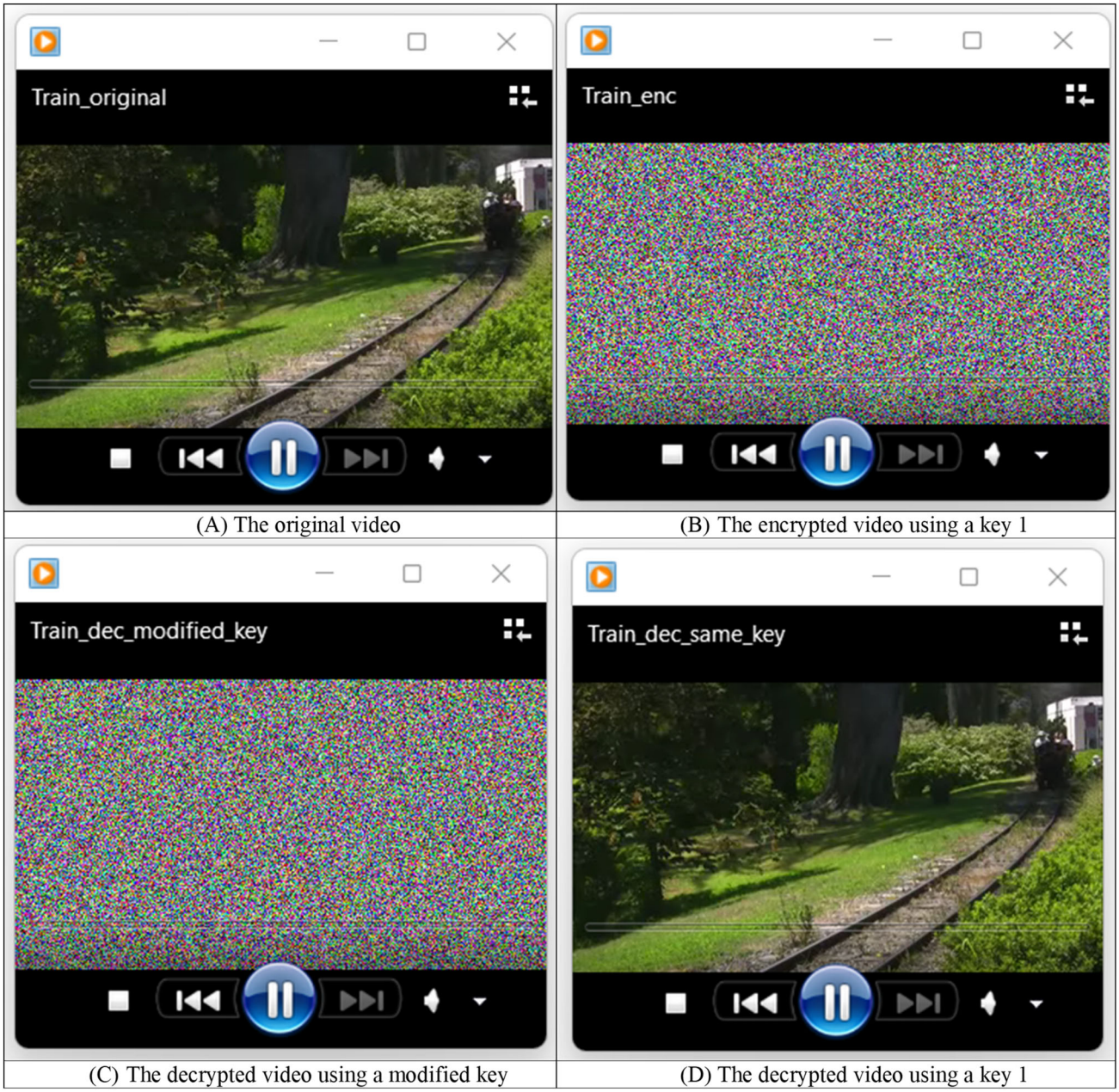


Fig. 14 Laplacian of Gaussian edge detection results of the original, encrypted, and decrypted frame number 10 for various videos



(A) The original video (B) The encrypted video using a key 1

(C) The decrypted video using a modified key (D) The decrypted video using a key 1

Fig. 15 Sensitivity of the key for Train video

3.12.2 Gaussian noise

This type of noise occurs due to the limitation of the sensor during the acquisition of the video frames under low-light conditions. In this experiment, the Gaussian noise with a variance value of 0.005 is added to various encrypted video frames, and the decryption process is performed. Figure 17 shows that the decrypted videos are still intelligible, despite the effect of the noise on the video frames.

3.13 Occlusion attack analysis

This section clarifies the decryption capability of the proposed scheme during the transmission of an encrypted video in case part of it has been dropped or lost. The

experiment proves that the proposed scheme can resist the occlusion attack. Figure 18 shows the occlusion attacks on the various encrypted video frames and the decrypted frames.

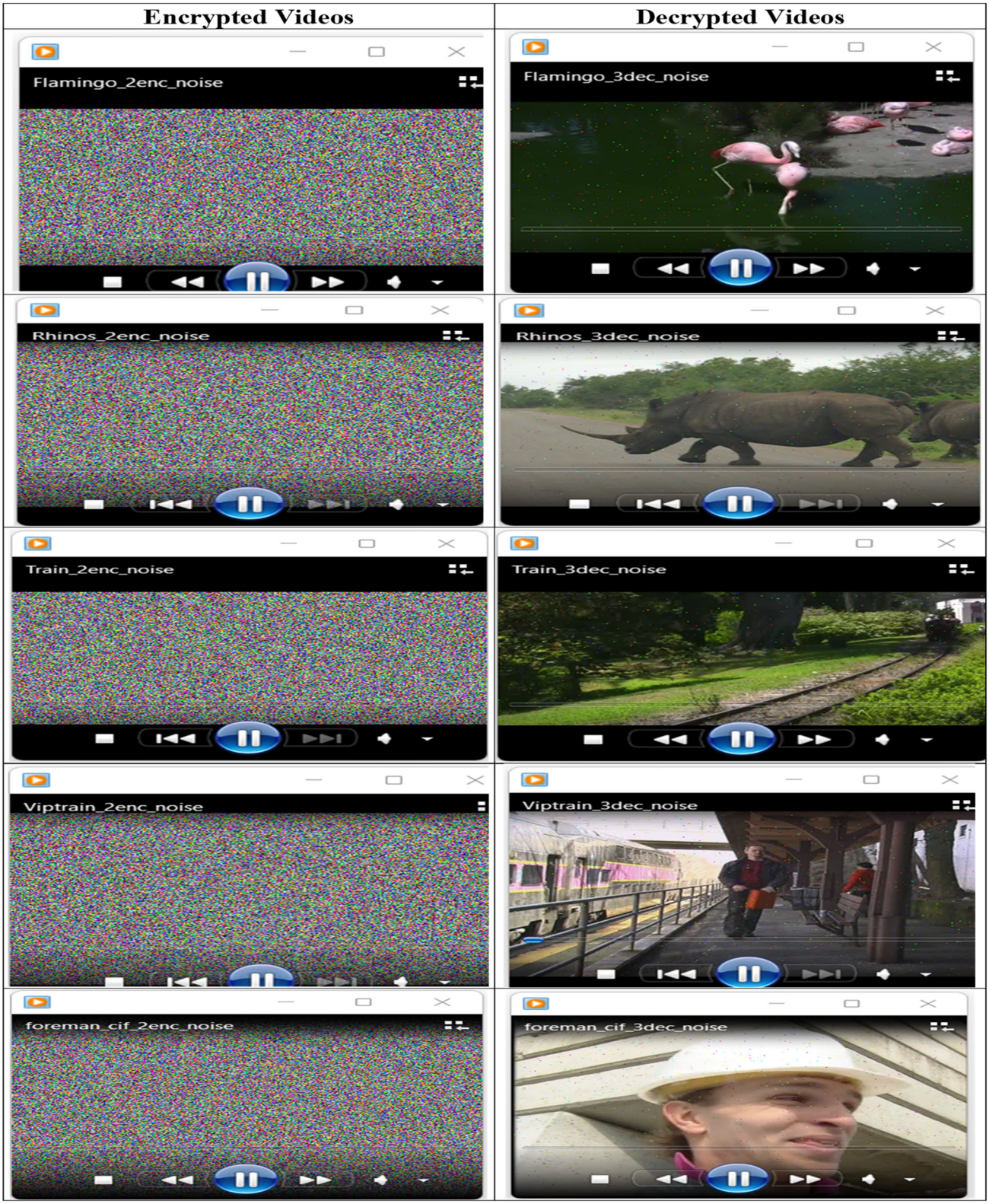


Fig. 16 Salt and pepper noise

Fig. 17 Gaussian noise

Fig. 18 Occlusion analysis

Table 8 EDR values for various videos

Video	EDR
Flamingo	0.9346
Rhinos	0.9218
Train	0.8683
Viptrain	0.8903
Foreman	0.9192

Table 9 The average execution time

Video	Duration (s)	Frame Rate	Video encryption time (s)
Flamingo	13.12	25	52.08
Rhinos	7.60	15	22.65
Train	10.48	25	41.82
Viptrain	20.87	30	136.76
Foreman	12.00	25	72.57

3.14 Execution time

The security scheme should encrypt/decrypt a colored video with low processing time. Various videos have been used to test the proposed scheme processing time. The experiment is carried out multiple times, and the average results are presented in Table 9. It is proven that the proposed scheme encrypts/decrypts the videos with high speed to fit the IoT devices' requirements.

3.15 Time complexity analysis

Each phase's complexity in the proposed scheme is computed, and then, the overall complexity for the proposed scheme is calculated. When the number of rows and columns is $M and N$ for the input video frame, $g = 2^n$ is the block dimension where $n = 4$. Therefore, the complexity for the channel splitting phase and scrambling phase is $O\left((M \times N)/g^2\right)$ and for the key generation and the channel diffusion phases is $O(M \times N)$. Then, $O(M \times N)$ is the overall complexity for one frame channel. Since, each frame has three channels, and each input video has several frames K. The $O(M \times N \times K)$ refers to the overall complexity of the proposed scheme.

Table 11 Execution time improvement ratio (ETIR)

Video	Ref. [13] (%)	Ref. [16] (%)
Rhinos	95.50321	98.03002
Viptrain	95.87074	98.08013
Foreman	95.63758	98.16124

3.16 Comparison with existing methods

A comparison between the proposed scheme and other recent encryption schemes is conducted to test the efficiency of the proposed scheme. The metrics used in this experiment are time complexity, execution time, execution time improvement ratio, correlation coefficient, NPCR, UACI, and entropy values.

The existing schemes have been implemented and executed in the same environment. Table 10 shows the time complexity and the average running time of 20 frames for various videos for the proposed scheme compared to the methods in [13, 16]. Table 11 presents the proposed scheme's execution time improvement ratio (ETIR) [44]. Also, Fig. 19 shows a visualized execution time. From the results, it is proven that the proposed scheme is faster than the methods in [13, 16], reflecting the proposed scheme's power.

Table 12 presents the average values of the correlation coefficient between adjacent pixels in the horizontal, vertical, and diagonal directions for the proposed scheme compared to the methods in [19, 35, 36, 45, 46] applied on Flamingo, Rhinos, Train, and Viptrain videos. The table shows that the proposed scheme has correlation coefficient values closer to zero than the mentioned works. Also, Table 13 presents the average values of adjacent pixels in the horizontal, vertical, and diagonal directions for the proposed scheme compared to the methods in [47–50] applied to Foreman video. The table shows that the proposed scheme has correlation coefficient values close to zero with the mentioned works.

Table 14 presents the average values of NPCR and UACI for the proposed scheme compared to the methods in [19, 35, 36, 45, 46] applied to Flamingo, Rhinos, Train, and Viptrain videos. Also, Table 15 presents the average values of NPCR

Table 10 Speed comparison

	Proposed method	Ref. [13]	Ref. [16]
Time complexity	$O(M \times N \times K)$ (s)	$O(M \times N \times K)$ (s)	$O(M \times N \times K)$ (s)
Rhinos	0.21	4.67	10.66
Viptrain	0.23	5.57	11.98
Foreman	0.26	5.96	14.14

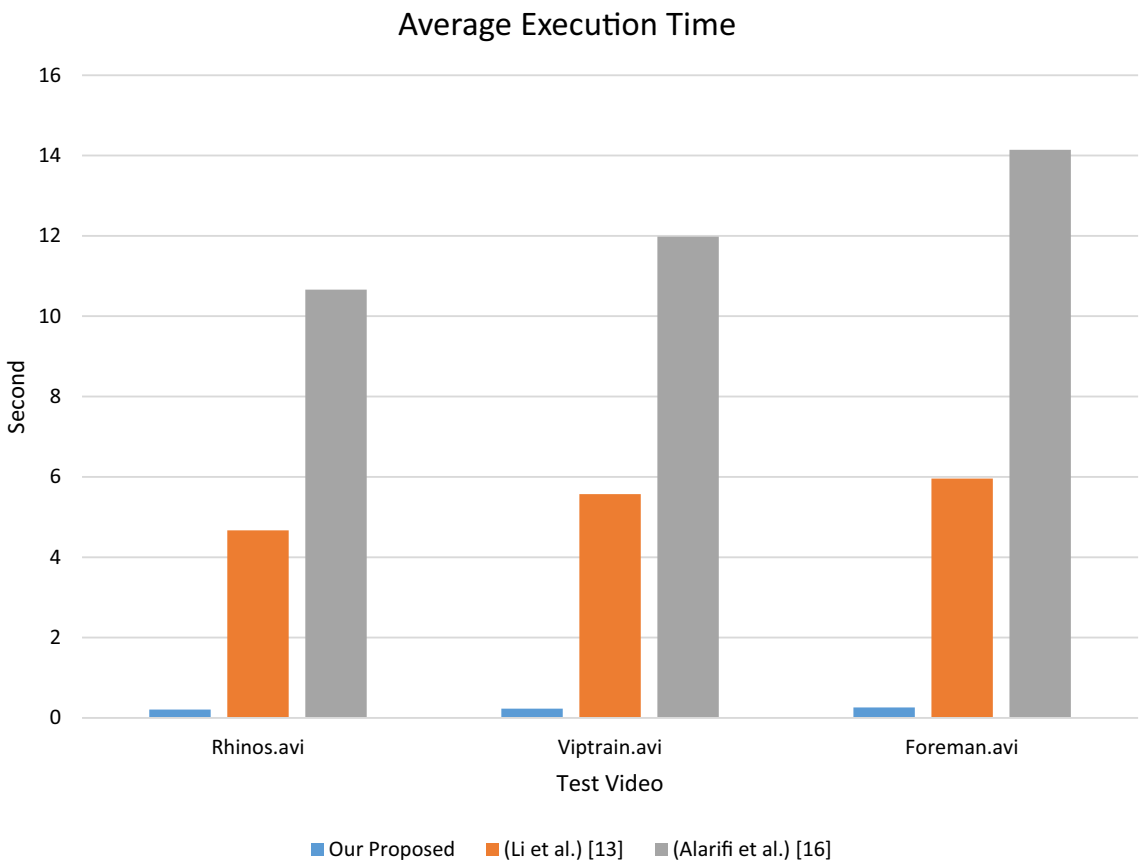


Fig. 19 Visual execution time

and UACI for the proposed scheme compared to the methods in [47–50] applied to the Foreman video. The results show that the proposed scheme has NPCR and UACI values closer to the NPCR and UACI optimal values than the other related works.

Table 16 presents the average values of Entropy for the proposed scheme compared to the methods in [19, 35, 36, 45, 46] applied to Flamingo, Rhinos, Train, and Viptrain videos. Also, Table 17 presents the average values of entropy for the proposed scheme compared to the methods in [47–50] applied in the Foreman video. The results show that the proposed scheme has entropy values close to the optimal value compared to the other related works.

4 Conclusion

This paper proposes a new scheme for securing the colored videos based on a frame channel scrambling and multi-key generation from a chaotic map. The proposed scheme is conducted independently on each of the three channels of the video frame to increase security. The performance of the proposed scheme is evaluated using visual analysis, histogram, correlation, entropy, differential attack, encryption quality analysis, PSNR, SSIM, and FSIM analysis, chosen-plaintext and known-plaintext attacks analysis, edges detection, keyspace, key sensitivity, channel noise attack

Table 12 Average correlation coefficient comparison applied on Flamingo, Rhinos, Train, and Viptrain videos

Algorithm	Video	*H*	*V*	*D*
Proposed scheme	Flamingo	0.0027	− 0.0034	0.0086
	Rhinos	− 0.0019	− 0.0130	− 0.0047
	Train	− 0.0065	0.0043	0.0098
	Viptrain	0.0089	− 0.0054	− 0.0151
Ref. [19]	Flamingo	Not specified	Not specified	Not specified
	Rhinos	Not specified	Not specified	Not specified
	Train	Not specified	Not specified	Not specified
	Viptrain	Not specified	Not specified	Not specified
Ref. [35] 12D Map	Flamingo	0.0155	0.0146	0.0171
	Rhinos	0.0196	0.0192	0.0138
	Train	0.0159	0.0195	0.0191
	Viptrain	0.0203	0.0175	0.0224
Ref. [35] Ikeda DDE	Flamingo	0.0150	0.014	0.0148
	Rhinos	0.0181	0.0140	0.0107
	Train	0.0154	0.0105	0.0121
	Viptrain	0.0176	0.0147	0.0158
Ref. [35] 8D Map	Flamingo	0.0227	0.015	0.0159
	Rhinos	0.0254	0.0184	0.0135
	Train	0.0186	0.0179	0.0155
	Viptrain	0.0192	0.0133	0.0165
Ref. . [36]	Flamingo	0.0150	0.014	0.0148
	Rhinos	0.0181	0.0140	0.0107
	Train	0.0154	0.0105	0.0121
	Viptrain	0.0176	0.0147	0.0158
Ref. [45]	Flamingo	0.0074	0.0061	0.0094
	Rhinos	0.0040	0.0080	0.0127
	Train	0.0063	0.0154	0.0126
	Viptrain	0.0020	0.0048	0.0114
Ref. [46]	Flamingo	Not specified	Not specified	Not specified
	Rhinos	Not specified	Not specified	Not specified
	Train	Not specified	Not specified	Not specified
	Viptrain	Not specified	Not specified	Not specified

Table 13 Average correlation coefficient comparison applied on Foreman video

Algorithm	Foreman		
	H	*V*	*D*
Proposed scheme	− 0.0099	− 0.0043	0.0116
Ref. [47]	0.0056	2.28e−5	0.0091
Ref. [48]	Not specified	Not specified	Not specified
Ref. [49]	0.0019	0.0045	0.0033
Ref. [50]	−0.0045	−0.0026	−0.0081

Table 14 NPCR and UACI comparison applied on Flamingo, Rhinos, Train, and Viptrain videos

Algorithm	Flamingo		Rhinos		Train		Viptrain	
	NPCR	UACI	NPCR	UACI	NPCR	UACI	NPCR	UACI
Proposed scheme	99.61	33.47	99.61	33.47	99.61	33.46	99.61	33.46
Ref. [19]	99.63	33.63	99.61	33.61	99.63	33.63	99.61	33.60
Ref. [35] 12D Map	99.37	33.49	99.44	33.49	99.36	33.49	99.34	33.46
Ref. [35] Ikeda DDE	99.94	33.59	99.95	33.59	99.94	33.50	99.95	33.59
Ref. [35] 8D Map	99.36	33.41	99.32	33.44	99.36	33.46	99.28	33.44
Ref. [36]	99.52	33.52	99.51	33.54	99.61	33.50	99.58	33.62
Ref. [45]	99.62	33.50	99.64	33.50	99.62	33.49	99.63	33.46
Ref. [46]	Not specified	Not specified	99.16	34.57	Not specified	Not specified	Not specified	Not specified

Table 15 NPCR and UACI comparison applied to Foreman video

Algorithm	Foreman	
	NPCR	UACI
Proposed scheme	99.61	33.46
Ref. [47]	Not specified	Not specified
Ref. [48]	Not specified	Not specified
Ref. [49]	98.63	34.03
Ref. [50]	99.86	33.56

Table 16 Entropy comparison applied on Flamingo, Rhinos, Train, and Viptrain videos

Algorithm	Flamingo	Rhinos	Train	Viptrain
Proposed scheme	7.99	7.99	7.99	7.99
Ref. [19]	Not specified	Not specified	Not specified	Not specified
Ref. [35]	Not specified	Not specified	Not specified	Not specified
Ref. [36]	Not specified	Not specified	Not specified	Not specified
Ref. [45]	7.99	7.99	7.99	7.99
Ref. [46]	Not specified	Not specified	Not specified	Not specified

analysis, occlusion attack analysis, computational processing time, and time complexity. The results proved that the proposed scheme is efficient in encrypting colored videos at high speed, does not require high computation resources, and is suitable for IoT devices. The proposed scheme is compared to the preceding related works, and the experiments prove that the proposed scheme has a high quality in securing the colored videos.

Table 17 Entropy comparison applied on Foreman video

Algorithm	Foreman
Proposed scheme	7.99
Ref. [47]	7.73
Ref. [48]	7.98
Ref. [49]	Not specified
Ref. [50]	7.99

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). No fund is available for this study.

Declarations

Conflict of interest The authors have no conflict of interest.

References

1. Von Solms, R., Van Niekerk, J.: From information security to cyber security. Comput. Secur. **38**, 97–102 (2013). https://doi.org/10.1016/J.COSE.2013.04.004
2. Liu, F., Koenig, H.: A survey of video encryption algorithms. Comput. Security. **29**(1), 3–15 (2010). https://doi.org/10.1016/J.COSE.2009.06.004
3. El-Shafai, W., Mesrega, A.K., Ahmed, H.E., Abdelwahab, N., AbdEl-Samie, F.E.: An efficient multimedia compression-encryption scheme using Latin squares for securing internet of things networks. J. Inf. Secur. Appl. **64**, 103039 (2022). https://doi.org/10.1016/J.JISA.2021.103039
4. Huang, X., Arnold, D., Fang, T., Saniie, J.: A chaotic-based encryption/decryption system for secure video transmission. IEEE Int. Confer. Electro Inf. Technol. **2021**, 369–373 (2021). https://doi.org/10.1109/EIT51626.2021.9491868
5. Liu, Y., Liu, S., Wang, Y., Zhao, H., Liu, S.: Video steganography: a review. Neurocomputing **335**, 238–250 (2019). https://doi.org/10.1016/J.NEUCOM.2018.09.091
6. Balaji, R., Naveen, G.: Secure data transmission using video Steganography. IEEE Int. Confer. Electro Inf. Technol. (2011). https://doi.org/10.1109/EIT.2011.5978601
7. Mustafa, R.J., Elleithy, K.M.: A highly secure video steganography using Hamming code (7, 4). IEEE Long Isl. Syst. Appl. Technol. Confer. LISAT **2014**, 2014 (2014). https://doi.org/10.1109/LISAT.2014.6845191
8. Dixit, M., Bhide, N., Khankhoje, S., Ukarande, R.: Video steganography. In: 2015 Int. Conf. Pervasive Comput. Adv. Commun. Technol. Appl. Soc. ICPC 2015 (2015). https://doi.org/10.1109/PERVASIVE.2015.7087159
9. Gupta, G., Gupta, V.K., Chandra, M.: An efficient video watermarking based security model. Microsyst. Technol. **24**(6), 2539–2548 (2018). https://doi.org/10.1007/S00542-017-3689-X/FIGURES/8
10. Asikuzzaman, M., Pickering, M.R.: An overview of digital video watermarking. IEEE Trans. Circuits Syst. Video Technol. **28**(9), 2131–2153 (2018). https://doi.org/10.1109/TCSVT.2017.2712162
11. JarJar, A.: Two Feistel rounds in image cryptography acting at the nucleotide level exploiting DNA and RNA property. SN Appl. Sci. **1**(11), 1–17 (2019). https://doi.org/10.1007/S42452-019-1305-7/TABLES/9
12. Lian, S.: Multimedia Content Encryption: Techniques and Applications. CRC Press (2008)
13. Li, X., Yu, H., Zhang, H., Jin, X., Sun, H., Liu, J.: Video encryption based on hyperchaotic system. Multimed. Tools Appl. **79**(33–34), 23995–24011 (2020). https://doi.org/10.1007/S11042-020-09200-1
14. Hosny, K.M., Kamal, S.T., Darwish, M.M.: Novel encryption for color images using fractional-order hyperchaotic system. J. Ambient Intell. Hum. Comput. **2022**, 1–16 (2022). https://doi.org/10.1007/S12652-021-03675-Y
15. Hosny, K.M., Kamal, S.T., Darwish, M.M., Papakostas, G.A.: New image encryption algorithm using hyperchaotic system and fibonacci Q-matrix. Electron **10**(9), 1066 (2021). https://doi.org/10.3390/ELECTRONICS10091066
16. Alarifi, A., Sankar, S., Altameem, T., Jithin, K.C., Amoon, M., El-Shafai, W.: A novel hybrid cryptosystem for secure streaming of high-efficiency H.265 compressed videos in IoT multimedia applications. IEEE Access **8**, 128548–128573 (2020). https://doi.org/10.1109/ACCESS.2020.3008644
17. Hosny, K.M., Kamal, S.T., Darwish, M.M.: A novel color image encryption based on fractional shifted Gegenbauer moments and 2D logistic-sine map. Vis. Comput. **2022**, 1–18 (2022). https://doi.org/10.1007/S00371-021-02382-1
18. Kaur, G., Agarwal, R., Patidar, V.: Color image encryption scheme based on fractional Hartley transform and chaotic substitution–permutation. Vis. Comput. (2021). https://doi.org/10.1007/S00371-021-02066-W/TABLES/8
19. Yasser, I., Mohamed, M.A., Samra, A.S., Khalifa, F.: A chaotic-based encryption/decryption framework for secure multimedia communications. Entropy **22**(11), 1253 (2020). https://doi.org/10.3390/E22111253
20. Munagala, V., Kodati, S.P.: Enhanced hole entropy-based encoding via whale optimization for highly efficient video coding. Vis. Comput. **37**(8), 2173–2194 (2021). https://doi.org/10.1007/S00371-020-01978-3/FIGURES/9
21. Faragallah, O.S., Sallam, A.I., El-Sayed, H.S.: Visual Protection Using RC5 Selective Encryption in Telemedicine. Intell. Autom. Soft Comput. **31**(1), 177–190 (2022). https://doi.org/10.32604/IASC.2022.019348
22. Dolati, N., Beheshti, A., Azadegan, H.: A selective encryption for H.264/AVC videos based on scrambling. Multimed. Tools Appl. **80**(2), 2319–2338 (2021). https://doi.org/10.1007/S11042-020-09654-3/TABLES/6
23. Faragallah, O.S., et al.: Cybersecurity framework of hybrid watermarking and selective encryption for secure HEVC communication. J. Ambient Intell. Humanize. Comput. (2021). https://doi.org/10.1007/S12652-020-02832-Z/TABLES/17
24. Cai, H., Sun, J., Gao, Z., Zhang, H.: A novel multi-wing chaotic system with FPGA implementation and application in image encryption. J. Real-Time Image Process. **2022**, 1–16 (2022). https://doi.org/10.1007/S11554-022-01220-4

25. Wu, X., Wang, K., Wang, X., Kan, H., Kurths, J.: Color image DNA encryption using NCA map-based CML and one-time keys. Signal Process. **148**, 272–287 (2018). https://doi.org/10.1016/J.SIGPRO.2018.02.028
26. Chai, X., Gan, Z., Yang, K., Chen, Y., Liu, X.: An image encryption algorithm based on the memristive hyperchaotic system, cellular automata, and DNA sequence operations. Signal Process. Image Commun. **52**, 6–19 (2017). https://doi.org/10.1016/J.IMAGE.2016.12.007
27. Wang, X.Y., Zhang, H.L., Bao, X.M.: Color image encryption scheme using CML and DNA sequence operations. Biosystems **144**, 18–26 (2016). https://doi.org/10.1016/J.BIOSYSTEMS.2016.03.011
28. Niyat, A.Y., Moattar, M.H., Torshiz, M.N.: Color image encryption based on hybrid hyper-chaotic system and cellular automata. Opt. Lasers Eng. **90**, 225–237 (2017). https://doi.org/10.1016/J.OPTLASENG.2016.10.019
29. Faraoun, K.M.: Fast encryption of RGB color digital images using a tweakable cellular automaton based schema. Opt. Laser Technol. **64**, 145–155 (2014). https://doi.org/10.1016/J.OPTLASTEC.2014.05.012
30. Wang, X., Zhang, H.L.: A color image encryption with heterogeneous bit-permutation and correlated chaos. Opt. Commun. **342**, 51–60 (2015). https://doi.org/10.1016/J.OPTCOM.2014.12.043
31. Auyporn, W., Vongpradhip, S.: A robust image encryption method based on bit plane decomposition and multiple chaotic maps. Int. J. Signal Process. Syst. (2014) https://doi.org/10.12720/IJSPS.3.1.8-13
32. Murali, P., Niranjana, G., Paul, A.J., Muthu, J.S.: Domain-flexible selective image encryption based on genetic operations and chaotic maps. Vis. Comput. (2022). https://doi.org/10.1007/S00371-021-02384-Z/TABLES/13
33. Yang, Z., Cao, Y., Ji, Y., Liu, Z., Chen, H.: Securing color image by using bit-level modified integer nonlinear coupled chaos model in Fresnel diffraction domains. Opt. Lasers Eng. **152**, 106969 (2022). https://doi.org/10.1016/J.OPTLASENG.2022.106969
34. Fang, P., Liu, H., Wu, C., Liu, M.: A survey of image encryption algorithms based on chaotic system. Vis. Comput. **2022**, 1–29 (2022). https://doi.org/10.1007/S00371-022-02459-5
35. Valli, D., Ganesan, K.: "Chaos based video encryption using maps and Ikeda time delay system," *Eur*. Phys. J. Plus (2017). https://doi.org/10.1140/EPJP/I2017-11819-7
36. Ranjith Kumar, R., Ganeshkumar, D., Suresh, A., Manigandan, K.: A new one round video encryption scheme based on 1D chaotic maps. In: 2019 5th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2019, pp. 439–444 (2019). https://doi.org/10.1109/ICACCS.2019.8728443
37. Song, X.H., Wang, H.Q., Venegas-Andraca, S.E., Abd El-Latif, A.A.: Quantum video encryption based on qubit-planes controlled-XOR operations and improved logistic map. Phys. A Stat. Mech. Appl. **537**, 122660 (2020). https://doi.org/10.1016/J.PHYSA.2019.122660
38. Ye, G., Pan, C., Dong, Y., Jiao, K., Huang, X.: A novel multi-image visually meaningful encryption algorithm based on compressive sensing and Schur decomposition. Trans. Emerg. Telecommun. Technol. (2021). https://doi.org/10.1002/ETT.4071
39. Duan, C.F., Zhou, J., Gong, L.H., Wu, J.Y., Zhou, N.R.: New color image encryption scheme based on multi-parameter fractional discrete Tchebyshev moments and nonlinear fractal permutation method. Opt. Lasers Eng. **150**, 106881 (2022). https://doi.org/10.1016/J.OPTLASENG.2021.106881
40. Gong, L.H., Luo, H.X., Wu, R.Q., Zhou, N.R.: New 4D chaotic system with hidden attractors and self-excited attractors and its application in image encryption based on RNG. Phys. A Stat. Mech. Appl. **591**, 126793 (2022). https://doi.org/10.1016/J.PHYSA.2021.126793
41. Kaur, G., Agarwal, R., Patidar, V.: Color image encryption scheme based on fractional Hartley transform and chaotic substitution–permutation. Vis. Comput. **38**(3), 1027–1050 (2022). https://doi.org/10.1007/S00371-021-02066-W/TABLES/8
42. YUV Sequences: http://trace.eas.asu.edu/yuv/. Accessed 07 Jan 2022
43. Kamal, S.T., Hosny, K.M., Elgindy, T.M., Darwish, M.M., Fouda, M.M.: A new image encryption algorithm for grey and color medical images. IEEE Access **9**, 37855–37865 (2021). https://doi.org/10.1109/ACCESS.2021.3063237
44. Hosny, K.M.: Fast computation of accurate Zernike moments. J. Real-Time Image Process. **3**(1–2), 97–107 (2008). https://doi.org/10.1007/S11554-007-0058-5
45. Sethi, J., Bhaumik, J., Chowdhury, A.S.: Chaos-Based Uncompressed Frame Level Video Encryption. Springer (2022). Accessed 15 Jun 2022 [Online]. https://doi.org/10.1007/978-981-16-6890-6_15
46. Elkamchouchi, H., Salama, W.M., Abouelseoud, Y.: New video encryption schemes based on chaotic maps. Wiley Online Libr. **14**(2), 397–406 (2019). https://doi.org/10.1049/iet-ipr.2018.5250
47. Kotel, S., Zeghid, M., Baganne, A., Saidani, T., Daradkeh, Y.I., Rached, T.: Fpga-based real-time implementation of aes algorithm for video encryption. Recent Adv. Telecommun Informatics Edu Technol 27–36 (2014)
48. Cheng, S., Wang, L., Ao, N., Han, Q.: A Selective Video Encryption Scheme Based on Coding Characteristics. Symmetry **12**(3), 332 (2020). https://doi.org/10.3390/SYM12030332
49. Elkamchouchi, H., Salama, W.M., Abouelseoud, Y.: New video encryption schemes based on chaotic maps. IET Image Process. **14**(2), 397–406 (2020). https://doi.org/10.1049/IET-IPR.2018.5250
50. Hafsa, A., Fradi, M., Sghaier, A., Malek, J., Machhout, M.: Real-time video security system using chaos- improved advanced encryption standard (IAES). Multimed. Tools Appl. (2021). https://doi.org/10.1007/S11042-021-11668-4/TABLES/14

Khalid M. Hosny was born in 1966, Zagazig, Egypt. He is a professor of information technology at the Faculty of Computers and Informatics at Zagazig University. Dr. Hosny received the B.Sc., M.Sc., and Ph.D. from Zagazig University, Egypt, in 1988, 1994, and 2000. From 1997 to 1999, he was a visiting scholar at University of Michigan, Ann Arbor, and the University of Cincinnati, Cincinnati, USA. Dr. Hosny is a senior member of ACM and IEEE; he is an editor and scientific reviewer for more than 50 international journals. His research interests include image processing, pattern recognition, multimedia, and computer vision. Dr. Hosny published three edited books and more than 100 papers in international journals. According to the recent edition of the Stanford rank, Dr. Hosny is included in the list of top 2% scientists worldwide.

Mohamed A. Zaki received the B.Sc. degree from the Department of information technology, Faculty of Computer and Informatics, Zagazig University, Egypt, in 2017, where he is currently pursuing a master's degree. His research interests include multimedia, cryptography, and image processing.

Nabil A. Lashin has received his B.Sc. in Communication and Electronics Engineering (1993) from Zagazig University, M.Sc. in Communication and Electronics Engineering (1999) from Cairo University, and his Ph.D. in Electrical Engineering and Computer Science (2005) from the Technical University of Berlin, Germany. He is currently an Associate Professor of Information Technology, with Zagazig University, Egypt.

Hanaa M. Hamza has received her B.Sc. in Computer Sciences (2006) from Zagazig University, M.Sc. in Computer Sciences (2010) from Zagazig University, and her PhD in Information Technology (2017) from Zagazig University, Germany. She is currently an Assistant Professor of Information Technology, with Zagazig University, Egypt.